



Enhancing robotic steel prefabrication with semantic digital twins driven by established industry standards

Lukas Kirner^{*}, Victoria Jung, Jyrki Oraskari, Sigrid Brell-Cokcan

Chair of Individualized Production (IP), RWTH Aachen University, Campus-Boulevard 30, Aachen, 52074, Germany

ARTICLE INFO

Keywords:

Semantic Digital Twin
Robotic steel prefabrication
DSTV-NC
Steel construction
Linked data
Semantic web

ABSTRACT

To increase automation in steel construction, new approaches are needed to strengthen the robustness of robotic steel prefabrication processes against manufacturing tolerances. While Digital Twins (DTs) can enable the detection of deviations and the adaptation of machine control accordingly, an adaptive information model interface that can integrate cross-process and cross-machine considerations is missing. Therefore, this paper focuses on the development of an ontology based on an existing steel prefabrication standard to link process data, tolerances and deviations, thereby enabling the realisation of a semantic DT. The approach's feasibility is proven by a case study that demonstrates process and tolerance modelling, as well as linking robot control, feedback data, and measured tolerances for robotic plasma cutting. The results show that by means of the resulting data model, a semantic DT can be realised, which allows making deviations and process knowledge available for downstream manufacturing, assembly and construction processes.

1. Introduction

In contemporary construction practices, partially automated production systems are predominantly applied for prefabrication processes. Though the degree of prefabrication is comparably high in steel construction, several characteristic traits hinder the integration of automated production systems from progressing further [1]. Firstly, unlike other industries, steel construction must manage higher component and manufacturer tolerances [2–5]. These occur due to geometric tolerances, material variations, irregular component sizes and weight, and small batch sizes of complex assemblies. Secondly, no continuous process information recording is usually established, nor does any standardised procedure for logging deviations exist. Thirdly, no adaptive information models are available for steel construction that can map an actual component's deviations. These traits prevent robotic applications in steel construction from being widely adopted since robotic path planning is traditionally based on ideal component dimensions [6].

Currently, some machines can already measure the exact dimensions of incoming material, adapt their path planning accordingly and finally capture the individual parts produced for quality assurance creating a bi-directional DT system [7]. Unfortunately, these machines only process and use information internally, acting as “data silos”. There is no defined method for feeding back into an information model that can be passed on to subsequent processes. This would be particularly desirable in assembly processes for multi-part assemblies,

as accessible tolerance information would allow updated path planning or geometric adjustments of dependent parts to ensure accuracy and efficiency. In robotic assembly, in particular, having accurate as-built information about the parts is needed to adjust robot paths and machine settings to reduce the risk of tool or machine damage due to unforeseen deviations [8].

One approach that shows potential to address these challenges is the semantic DT [9–11]. It goes beyond a simple DT by incorporating semantic information - meaningful context and relationships - through the use of ontologies. Ontologies add semantics to a data model by structuring and formalising the conceptual knowledge of a domain. They define the meaning of terms, specify relationships between concepts, support reasoning and inference, facilitate interoperability, and preserve domain understanding. For robotic steel prefabrication, this could enable the interpretation of DT data by subsequent users and the ability to pass data downstream in the value stream using the functionalities of the Semantic Web stack [12]. In addition, process knowledge could be derived from the resulting graph, potentially enabling the optimisation of machine parameters.

To enable semantic DTs for robotic steel prefabrication, an ontology is missing that maps and links information about the robotic setup, process data such as machine instructions and tolerances, and data feedback such as quality measurements and deviations. Therefore, this

^{*} Corresponding author.

E-mail addresses: kirner@ip.rwth-aachen.de (L. Kirner), jung@ip.rwth-aachen.de (V. Jung), oraskari@ip.rwth-aachen.de (J. Oraskari), brell-cokcan@ip.rwth-aachen.de (S. Brell-Cokcan).

<https://doi.org/10.1016/j.autcon.2024.105699>

Received 16 April 2024; Received in revised form 9 August 2024; Accepted 11 August 2024

Available online 25 August 2024

0926-5805/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

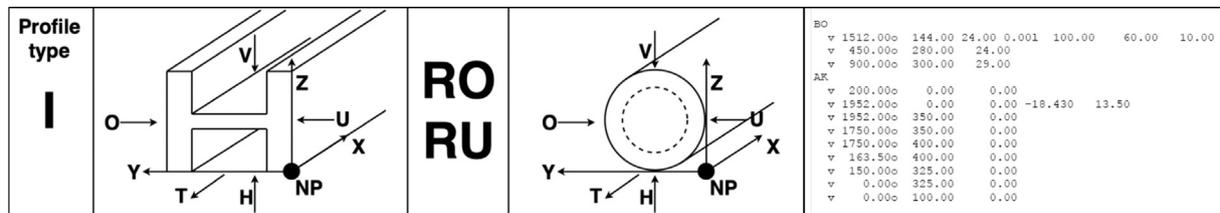


Fig. 1. DSTV-NC schemata of plane designations for different section types (left) and the common, ASCII-based notation (right) according to [13,14].

research focuses on developing such an ontology and demonstrating its capabilities using a case study involving a real robotic steel prefabrication process. In order to ensure the practicality and compatibility of the proposed solution with existing machine interfaces, the development will be based on an existing standard, the DSTV-NC (see Section 2.2), which will be explained in detail below.

2. Related work

2.1. DSTV-NC - A steel manufacturing standard

The *Deutscher Stahlbau-Verband Numerical Control* (DSTV-NC) standard in steel construction describes an interface between design, production planning, and machine control. It is a commonly used interface between computer-aided design and computer-aided manufacturing (CAD/CAM) applications and numerical control (NC) manufacturing. It provides efficient communication of data essential for fabricating and processing steel components, such as beams and plates [13,14]. The interface can transfer component information but primarily provides geometric manufacturing descriptions without detailed machining information. Fig. 1 shows an extract from the existing DSTV-NC schema for plane designations and the position of the coordinate system. The form elements of a part are assigned to the appropriate machining plane and positioned in the coordinate system of that plane.

The DSTV-NC standard allows various NC machines to be controlled via their respective post-processor. Subtractive manufacturing processes use saw drilling machines, flame-cutting and punching machines, and 5-axis computer numerical control (CNC) machining. One of the key benefits of DSTV-NC is its ability to standardise data exchange between different software applications and machines, reducing errors and improving interoperability across the supply chain.

The steel construction industry faces challenges due to a shortage of skilled workers and the need for digitalisation and automation. For this reason, [15] investigated whether the existing DSTV-NC format can be used to control robots that can carry out steel machining processes. A prototypical interface for controlling an industrial robot was implemented by developing an independent plugin for the visual programming tool Grasshopper for Rhinoceros 3D. Grasshopper, developed by David Rutten at Robert McNeel & Associates [16], provides an accessible visual programming environment, integrated visualisation tools, and modular extensibility through libraries and plugins. However, programming remains a significant barrier to using robots for single tasks and small batches, as knowledge and skills are required to adapt path planning to the task at hand. To overcome this problem, an efficient task-oriented approach was chosen to implement the interface, where a task corresponds to a DSTV-NC machining step. A strategy for these tasks was stored in the interface, and the robot code was automatically generated. Rule-based global path planning was developed and implemented for collision-free machining of multiple component sides [15].

During validation, it was highlighted that the DSTV-NC format does not contain actual part information as the information relates to the design geometry. Due to the specific characteristics of steel and the large part sizes, there are deviations from the planned geometry to the actual dimensions of the steel part. This information is currently

missing in the DSTV-NC format, so the path planning has to be adapted manually to avoid collisions between the robot and the part. The modular structure of this format could allow it to be extended with additional information. So far, it has only been used as a manufacturing instruction with ideal geometries, without the possibility of feedback on actual values or updating values [17].

The limitations of the DSTV-NC format become clear during validation. Although the format is standardised and straightforward in describing steel components, it has limitations in mapping complex components and assembly information. This information has been partially regulated through the BFS-RL-03-108 interface [18].

There is an increasing need for process and component-related information specifically required for automation in steel prefabrication use cases. While the approach presented in this paper focuses on implementing structured information and knowledge in the steel construction industry with help of Linked Data (LD) [19–21], an ad hoc group “NC” of the Working Committee “Information Technology” of *bauforumstahl/DSTV* currently focuses on transferring information from the DSTV-NC standard to Industry Foundation Classes (IFC), the standard exchange format for Building Information Modelling (BIM) data [22]. IFC Industry Foundation Classes (IFC, defined by ISO/PAS 16739 [23]) is the standard exchange format for Building Information Modelling (BIM). It could be an alternative for storing additional tolerance and process information to enable a robotic process for steel prefabrication. Rocha et al. [24] has demonstrated the benefits that can provide for the steel construction industry. IFC forms the basis for an open, neutral and standardised data format and, in addition to the 3D data model, contains extensive data structures for describing objects from almost all sectors [17,25].

Both LD and IFC approaches initially use the same basic principles, as it is first necessary to define which information is to be digitally mapped for the various processes in steel construction. Only then does the modelling take place. At this point, however, it can be mentioned that Linked Data and IFC are not mutually exclusive and that a converter already enables data exchange [26,27].

2.2. Robotic prefabrication in steel

In practical steel construction, robotics’s developments and potential have been poorly represented. In particular, using robots in assembly is hampered by high investment risks and limited applicability. Due to high tolerances, direct CAD/CAM control is not easily feasible [2]. Therefore, industrial robots require flexible process control through efficient and robust tolerance detection and processing. However, achieving the desired comparison requires time-consuming teaching or costly measurement and sensor technology with coordinated control [28–30]. The latter approach is becoming increasingly important in research and development. Several research projects have already explored solutions for small and medium-sized enterprises (SMEs). Projects such as “SMERobotics: Cognitive Robotic Systems for SMEs” [31], “IVGT: Intelligent Processing of Large Components with High Tolerances” [3,5,32] and “ROPROF: Robotic Welding of Individually Designed Steel Profiles in Small Batch Sizes” [33] focused on the development of sensor-supported robotic welding systems for customised steel profiles. The ROPROF and IVGT projects were explicitly

concerned with the automated welding of tolerance-sensitive steel components for low-volume construction [32,33]. Using sensors and control algorithms, both projects successfully welded tolerance-sensitive components in demonstrators [3,29].

The manufacturing process of steel construction is predominantly carried out in individual and small series production [34]. For the implementation of robotics in SMEs, it is crucial to minimise the technical effort and, hence, the investment cost while maximising the flexibility and adaptability of the system. Increased flexibility can be achieved, among other things, through easy programming of the manufacturing system, as can be derived from path planning based on CAD models [29].

Integrating robotics into steel prefabrication requires digital data that the robot can use to execute tasks based on actual workpiece dimensions. Ensuring a meaningful and informative representation of tolerances and effective tolerance management is equally important as implementing tolerance compensation methods. In addition to this data, providing further information and links to other processes and stakeholders and integrating process feedback for subsequent robotic processes is of importance and should be accessible within a unified data environment.

2.3. Linked Data for robotic manufacturing and construction

Ontologies have become a promising solution for tackling the issue of semantic interoperability [35–38]. They are formal specifications about concepts within a specific domain, frequently employing logical theory and reasoning mechanisms to derive novel insights [39]. By providing explicit data semantics, ontologies facilitate semantic interoperability by distinctly and unambiguously portraying entities, concepts, and their interrelations [40].

In recent years, many ontologies have emerged targeting the construction domain. The predominant approaches involve translating existing models [27,41,42] or developing novel mapping techniques [43]. Diverse research endeavours have focused on outlining general construction processes. One such ontology, the Internet of Construction (IoC), centres on the concept of *ioc:process*, to bridge disparate subdomains within the construction realm, including steel construction [44, 45]. Additionally, the ifcOWL ontologies [26,41] incorporate the *ifc:IfcTask* class, which has properties that support its association with construction components, subtasks, and resources. The LinkOnt extension augments terminology pertinent to task-level and resource information [46]. However, particular ontologies, such as Monument Damage Information System (MONDIS)[47] and the Conservation ProcessModel (CPM), are limited in scope and focus solely on inspection and repair. Conversely, the Construction Tasks Ontology (CTO) comprehensively depicts the tasks inherent in construction projects, encompassing installation, removal, modification, inspection, and repair. The tasks within CTO can be organised using *cto:TaskContext*, with preventive maintenance tasks classified as repair or modification [46].

The ifcOWL-DfMa ontology represents an extension of the ifcOWL ontology, explicitly aiming to translate terminology from the off-site construction domain into a machine-interpretable format. Aligned with the ifcOWL ontology, it is designed so that each *dfma:Building* corresponds to an *ifcBuilding*. 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: *DfMA_Production_Process* (encompassing production and ancillary activities), *Resources* (comprising labour, materials, overhead, and equipment), *Activities* (encompassing production tasks and associated resources) and *Modality* (including platform, time, location, and transportation aspects) [48].

El-Gohary and El-Diraby [49] introduced the IC-PRO-Onto, a domain ontology focusing on infrastructure and construction processes, which conceptualises knowledge process-oriented. For example, this

ontology encapsulates models of activity-related constraints. The Digital Construction Ontologies (DiCon) constitute a suite of ontologies delineating intricate construction workflows and their associated entities and relationships. DiCon encompasses six modules designed to specify domain knowledge within the construction domain, covering entities, processes, information, agents, variables, and contexts [43].

Lee et al. demonstrated that integrating an ontology model that takes into account relevant details about components, such as geometry, material properties and production speed, can ensure efficient information management to support on-site prefabrication and assembly processes. Their model elucidates the intricate interplay between components, materials, equipment and labourers involved in prefabricating concrete panels outside the site and the subsequent installation on the site [50].

This paper builds on and extends the application of ontologies in prefabrication within the construction sector. Lastra and Delamer advocate using ontologies to streamline engineering efforts and expedite the establishment of cost-effective manufacturing production systems. Consequently, ontologies are designed following a modular and reusable framework, enabling the encapsulation of specific knowledge within the manufacturing domain. Central to the developed ontology is the core concept of elucidating the relationships between the product, the manufacturing processes necessary for its production, and the equipment that facilitates it [51].

A primary advantage of ontologies lies in their capacity to incorporate well-established concepts, such as the DSTV-NC format, into the semantic web technology framework. This enables the integration of disparate and unstructured data sources, including but not limited to Building Information Modelling (BIM) or scheduling data. However, previous research [52,53] has focused predominantly on generic construction processes, overlooking the specificity of steel construction processes. Consequently, a gap exists in prior investigations concerning the formulation of domain-specific ontologies tailored to steel construction.

As a response to this need, prior efforts have culminated in developing the DSTV domain ontology, which is expounded below.

2.4. Semantic Digital Twins

The concept of the Digital Twin (DT) was initially introduced in the domain of Product Lifecycle Management (PLM) as the “Information Mirroring Model”[54]. It was further refined as “a set of virtual information constructs that fully describes a potential or actual physical manufactured product which “at its optimum, any information that could be obtained from inspecting a physical manufactured product can be obtained from its Digital Twin”[55]. It was quickly adopted by other domains, such as mechanical engineering or robotics, to describe machine and factory setups with a bidirectional data flow [56,57].

Semantic DTs represent a significant advancement of the DT concept by integrating semantic technologies to enhance data interoperability, contextual understanding, and process automation. Unlike traditional DTs, the approach leverages ontologies to imbue digital models with rich semantic metadata [9–11].

For a logistics use case, Kümpel et al. demonstrate how a semantic DT (here described as *semDT*) can be used to create a semantically enhanced virtual representation of a retail store environment. It connects a knowledge base incorporating various interchangeable knowledge sources with a scene graph and sensor data feedback. The authors show how enhancing the DT with semantic data can allow complex reasoning tasks for retail business as well as improved data exchange [58].

In Addition to the Semantic Web interfaces, research is also conducted on implementing common web interfaces [59]. An extensive overview of future directions for semantic DTs in construction is presented by Boje et al. [11].

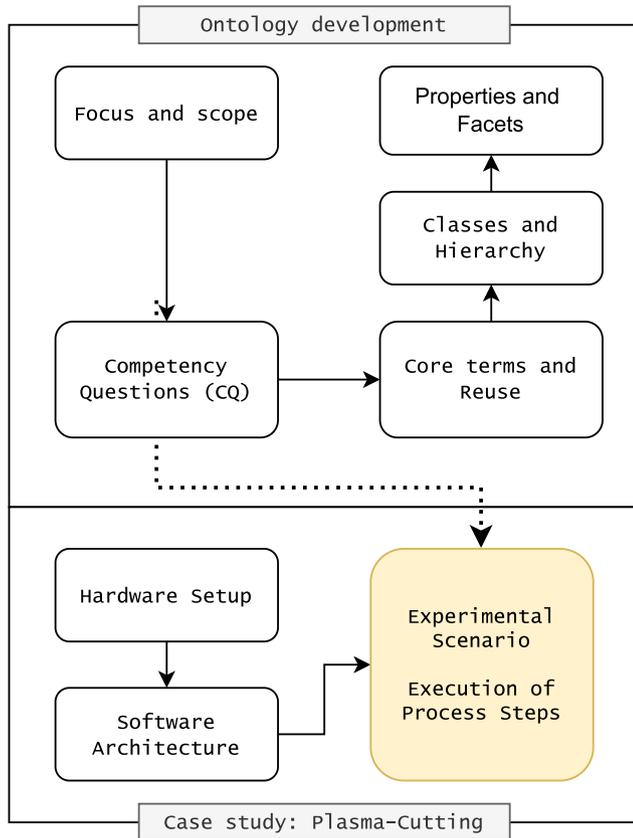


Fig. 2. Overview of methodological steps employed in this paper.

3. Methodology

As seen in Fig. 2, the methodology used in this paper consists of two main parts: the development of the ontology and its evaluation using a case study.

The **ontology development** follows several steps, starting with the description of its scope and the definition of a set of so-called “Competency Questions” (CQ), which serve to develop and evaluate the ontology in a practice-orientated way. Furthermore, the connection and reuse of existing ontologies are examined. After that, the core concepts of the resulting ontology are explained in detail. This is done using two examples. First, through-hole drilling is modelled, as it is a comparatively simple process well defined in the DSTV-NC standard (*hljob*, in the standard, a job to create a hole). This is then extended and compared with the semantic mapping of a more sophisticated method, which describes the plasma-cutting of a workpiece and involves using more complex Linked Data modelling techniques. Finally, extensions to the DSTV-NC standard described in the ontology for process and tolerance modelling are presented.

The **case study** is used to demonstrate how the developed ontology can be employed to create a semantic DT for a robotic plasma-cutting process. Although it shows the potential of Linked Data as a bridging element for existing information sources and frameworks, its primary goal is to evaluate the ontology to prove its practicality. To this end, a process flow and control logic are created and executed. The results will be checked against the CQs defined for the development of the ontology.

4. Ontology development

4.1. Focus and scope

As the schema for the XML-based DSTV-NC standard (*XNC*) is available [14], a direct translation to Resource Description Framework (RDF) [60] was initially chosen.

RDF is part of the Semantic Web technology stack introduced by Tim Berners-Lee in 2000. It defines information as directed graphs composed of triple statements (subject–predicate–object) representing the relationships and properties of resources within an ontology. RDF uses Uniform Resource Identifiers (URIs) to uniquely identify each resource, ensuring that every ontology concept, property, and instance is uniquely and universally recognisable. Its standardised format ensures that ontologies can be shared, understood and reused across different systems and applications, facilitating interoperability and data integration. [12,61]

The initial prototype was limited to workpiece information and exemplary drilling processes described as *hl* or *hljob* (as short for “hole”), resulting in a W3C Web Ontology Language (OWL) [62] definition with 258 classes and 1583 axioms. The generated ontology retained the original standard structure, simplifying conversion to and from the RDF-based version. However, even this incomplete fragment resulted in a complex OWL structure unintelligible to the human reader.

Consequently, the work presented in this paper focuses on creating the ontology in a simplified manner, adding only the existing concepts from the standard where they are needed while reusing existing ontologies as much as possible. The methodology we used for development is described in “Ontology Development 101: A Guide to Creating Your First Ontology” [63].

First, Noy and McGuinness advise clarifying the focus and scope of the ontology. This is done by defining and answering three Scope Questions (SCQ):

SCQ1 what domain should the ontology cover? The domain of steel construction and steel prefabrication

SCQ2 what is the purpose of the ontology? The ontology should allow the description of a process chain that can be used in steel construction and prefabrication. This chain also links measurement deviations, tolerances, and machine data. The ontology should help model the processes using terms from the DSTV-NC standard. This will allow concepts to be reused, easily mapped, and aligned with the standard’s existing file formats.

SCQ3 what kind of questions should the ontology be able to answer? The ontology should allow for the description of the production process, including the required data. This means it should answer questions about the planned data and, therefore, the basis for executing the method. It should also allow for the addition of measurement data, associated tolerances, and validation information to answer questions about adapting subsequent processes or quality control. It should be possible to optimise the production process by querying the resulting datasets. Furthermore, digital and Linked Data can later serve as a basis for making well-founded statements about material properties and assessing whether or not the components can be deconstructed and reused after the first life cycle.

4.2. Competency questions (CQ)

Competency questions were developed based on the scope specification (see Table 1). These CQs were derived from the previous content of the standard, the results of interviews and exchanges with BauFeSt project industry partners, and previous research results in Linked Data.

Competency questions are technical-functional. They describe precisely what queries should be able to answer once the ontology has been established. These queries are not limited to only using the concepts of

Table 1
Competency Questions (CQ)

| Type | No. | Question |
|--------------------|-----|---|
| Plan data | CQ1 | What NC instruction must I pass to the machine? |
| Measurement data | CQ2 | What are the measured values? |
| Tolerance data | CQ3 | What are the tolerances for the produced feature? |
| Validation data | CQ4 | Is the feature within the tolerance? |
| Adjustment | CQ5 | Do I need to adapt the following processes? |
| Resource | CQ6 | What machines and what tools are used? |
| Machine parameters | CQ7 | What are the process parameters used? |
| Process knowledge | CQ8 | Can process knowledge be deduced from the resulting data? |

Table 2
Overview of connected and aligned ontologies.

| Namespace | Main classes/focus and purpose of the ontology | Reference |
|-----------|---|-----------|
| bot | <i>bot:Zone, bot:Element, bot:Interface</i> An ontology that describes the core topological concepts of a building and the relationships between the concepts. One of the central ontologies introduced within the Linked Building Data (LBD) group. | [64] |
| ifc | <i>ifc:Root, ifc:ObjectDefinition, ifc:Relationship, ifc:PropertyDefinition</i> Web Ontology Language (OWL) representation of the Industry Foundation Classes (IFC) schema. | [65] |
| ioc | <i>ioc:Process, ioc:Element, ioc:Resource</i> An ontology that was developed to describe processes and metadata along the construction value chain. | [45,66] |
| opm | <i>opm:PropertyState</i> An ontology for describing properties that change over time. | [67] |
| schema | <i>schema:Thing</i> A collaborative project to develop schemas for structuring data. | [68] |

the developed ontology, but they can also show how the ontology can be combined with other vocabularies.

One of the principles of Linked Data is the reuse of existing ontologies. Our research on the state of the art of ontologies for steel construction showed that hardly any ontology approaches have been developed for this domain. The competency questions cannot be fully answered in any of the available solutions, especially in a practical way for robotic use. For concepts describing the building process, building elements, and element metadata, there are ontologies that we consider to be applicable and mature. By incorporating them, we aim to achieve greater interoperability. Table 2 provides an overview of reused or linked ontologies.

4.3. Ontology development

Instead of mechanically translating the standard into an ontology description, we crafted the new ontology to fully leverage the potential of the ontology and Linked Data approach. We employed a hole-drilling use case to enumerate essential terms in the ontology and adopted a similar methodology for plasma-cutting.

The core concepts derived from the DSTV-NC standard encompass features such as a trough hole, feature values including diameter and depth, the reference plane, tolerances such as depth tolerance, and reference views like the top and front views. New concepts inferred from CQ2, CQ3, and CQ4 include the measured values for diameter and depth and the calculated values for deviations in these measurements. Additionally, measuring, validation, production, and adjustment processes were incorporated to address CQ4, CQ5, and CQ8.

The Internet of Construction Process Ontology was considered and adapted for the process descriptions. The implementation was left user-defined for the units of measurements and values to keep the ontology modular. Ontologies that can be used for this are MUO[69], QUDV [70], OM [71], and QUDT [72]. Also, Custom Datatype can be used as described in [73].

We have followed the bottom-up approach to create the class hierarchy. First, classes for the specific terms were created following the camel case naming convention. Then, the concepts were grouped to find more general ones, avoiding creating generalised classes with low information content. The process resulted in top-level classes like Feature, FeatureValue, and Process. The design of classes and property slots was use-case and competency question-driven.

The full ontology description is available at <http://w3id.org/dstv/>.

4.4. DSTV concept descriptions and ontology structure

The DSTV ontology is intended to be a domain ontology in conjunction with the IoC project's top-level process ontology. Therefore, DSTV process classes, subclasses of *ioc:process*, are the foundation of the ontology. From the original standard, the *dstv:ProductionProcess* class can be derived as the basic production process description.

4.4.1. ThroughHole drilling example

To illustrate the structure of the resulting ontology, Fig. 3 shows an example of a simple *dstv:ThroughHoleDrill* (hljob) and the concepts associated with it. The DSTV-NC standard, as initially presented in the American Standard Code for Information Interchange (ASCII) specification, only covers the description of the incoming material (in this case, the "initial beam"), the description of the reference plane, and the description of the intended features. Relationships and concepts covering these in OWL were the first to be incorporated, as the minimum use of the ontology should be to describe the same content as a DSTV-NC file. We can model the resulting process feature (*dstv:ThroughHole*) by using the basic functionality of the *ioc:process* class to distinguish between element input and output. The feature can map to a corresponding *ifc:OWL* feature, in this case, the *ifc:VoidingFeature*, which is a predefined type *ifc:Hole*.

To place the process in a sequence (*ioc:hasPredecessor*, *ioc:hasSuccessor*) as well as to associate process metadata such as status or actor, additional concepts of the *ioc:Process* superclass can be used. A particular case is the *ioc:Resource*, which describes the machine in use for the process. The DSTV-NC standard defines a large amount of metadata associated with this concept. However, the expressiveness is limited because this metadata is in the file header. This means these NC files can only describe processes the same machine performs. A Linked Data approach can easily overcome this limitation. Metadata can be linked to a *dstv:Resource* class to be included in subsequent ontology iterations.

The main idea of extending the DSTV-NC standard to include data related to Measurements, Deviations and Tolerances is to find a clean structure to describe the data and how they relate. This is done by the main concept of *dstv:FeatureValues*. In the example shown in Fig. 3, one of these is of type *dstv:Diameter*. With the help of constraints in OWL, the model is defined so that each *dstv:ThoroughHoleDrill* has exactly one diameter. The diameter can be explicitly associated with

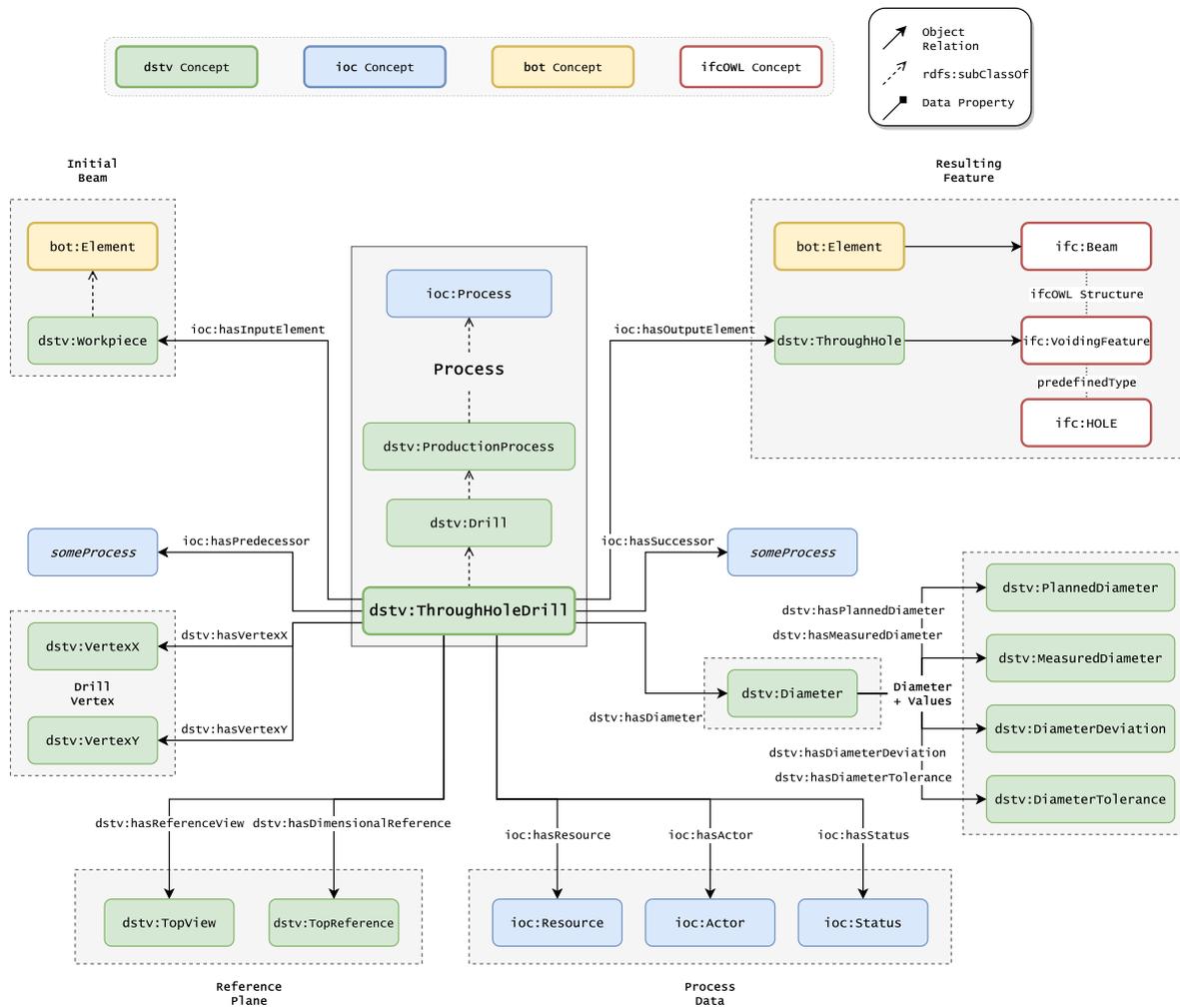


Fig. 3. Concepts of the DSTV ontology with *dstv:ThroughHoleDrill* as example.

values that specify in their relation and range classes whether they are planned or measured values or represent a deviation or tolerance. ASCII or XML-based NC file data is mapped to the planned values. The object property is defined for each subclass to implement validation via the W3C Shapes Constraint Language (SHACL) [74] and to keep the data human-readable. This has been done so that concepts from the Ontology for Property Management (OPM) [67] can be added to maintain a history for change management purposes.

4.4.2. Plasma-cutting example

As an extension of the drilling process example, we will present a more complex example of a plasma-cutting process. This is chosen as it contains the RDF lists required for all DSTV processes that involve paths or multiple sub-process steps.

In the DSTV-NC, plasma-cutting is classified as internal contouring, a subclass of the *dstv:ProductionProcess*. The concepts related to *dstv:InternalContouring* (see Fig. 4) are similar in structure to the drilling process shown in Fig. 3. Both use ioc concepts to describe a process chain and the *dstv:workpiece* as an input and output element. Other IoC ontology concepts work similarly to defining reference planes with *dstv:hasReferenceView* and *dstv:hasDimensionalReference*, but they are not displayed in Fig. 4 for the sake of readability. However, some differences require attention.

The process creates the *dstv:internalContour*, which can be linked to the corresponding *ifc:voidingFeature*. However, no unambiguous predefined type in the IFC data model can be cleanly used for further specification. It is important to note that a contour consists of several

dstv:Vertex in a specified sequential order. The class *dstv:FeatureValueList* is used to describe an RDF-list, with entries modelled via *list:hasContents* and *list:hasNext*. The vertices are divided into their x and y components (*dstv:VertexX* and *dstv:VertexY*). Planned and measured values, deviation, and tolerance can be linked at this level in the hierarchy, similar to the drilling example. In the case study on robotic plasma-cutting (see Section 5), the contouring process is connected to a methodical description of robot commands through the use of *ioc:hasMethod*. The case study section explains the commands and domain ontology created for this purpose.

4.5. Process extension

As previously stated, the DSTV-NC standard's existing processes describe production processes. The same logic used for *dstv:TroughHoleDrill* or *dstv:InternalContouring* applies to all these production processes. Three additional process types have been added to the DSTV ontology to take advantage of the concept's potential for automation.

For the measurement of a feature value, such as *dstv:InternalContour*, the *dstv:MeasurementProcess* class is employed. It adds a measured value to one or several feature values, which can be compared to a planned value. The class has further subclasses that correspond to each production process. In this example, the *dstv:InternalContouringMeasurement* class can be associated with *dstv:InternalContouring*. The *dstv:ValidationProcess* involves calculating *dstv:DeviationValues* and comparing them to aligned tolerances. Depending on the output, a

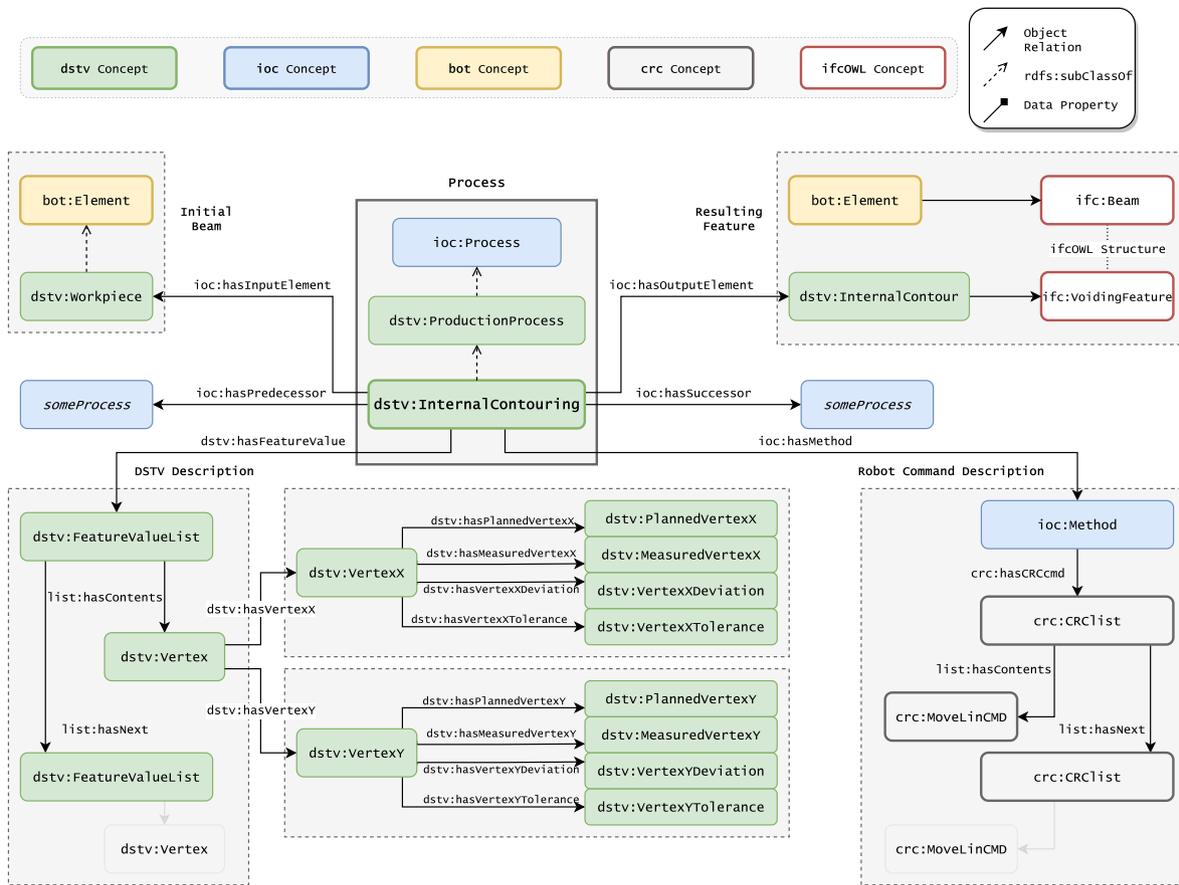


Fig. 4. Concepts of the DSTV ontology with *dstv:InternalContouring* as example.

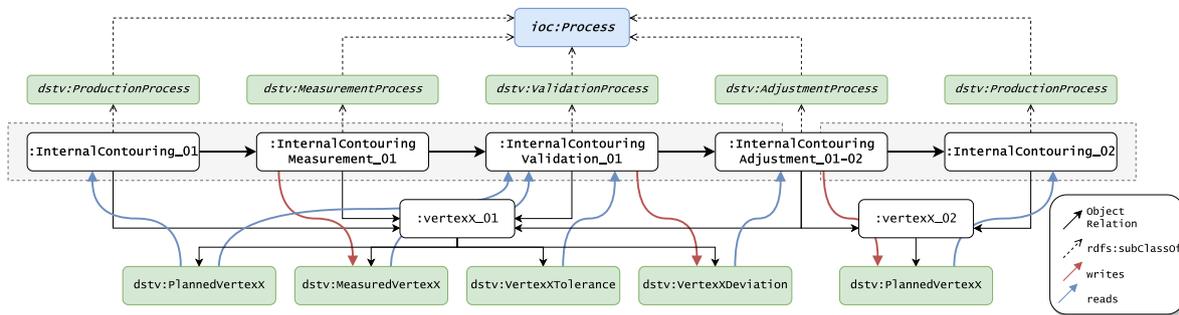


Fig. 5. Process-types in a sequential use-case for *dstv:InternalContouring*.

dstv:AdjustmentProcess can be executed to create a new set of planning data for the following production processes. The precise structuring of these processes is particularly beneficial when using a multi-robot cell to execute the entire workflow. Depending on the setup, these processes may be performed individually by different machines, algorithms, or workers, and they must be planned and scheduled to work together effectively. Fig. 5 illustrates the exemplary use of the extension in the case study described in Section 5, where all the aforementioned process types are used sequentially.

4.6. Tolerance extension

Tolerances are critical in defining the allowable deviations in the dimensional and geometric parameters of steel components, such as beams, columns, and joints. They help ensure that the various parts of a structure will fit together correctly when assembled to provide the desired functional and load-bearing performance. Extensive research

and personal discussions with experts in the steel construction industry have shown that some machines can already measure the part before and/or after it is machined automatically. However, users must manually enter the tolerance definitions because they are only available in human-readable files and printouts. As a result, the measurements cannot be validated against the tolerance definitions or stored.

Within our research, we investigated the manufacturing tolerances of various steel profiles and the linked tolerances. Manufacturing tolerances of steel sections refer to the acceptable deviations of the steel components, such as section height, flange width, and web or flange thickness. The tolerances for steel sections can be found in relevant building codes, standards, and specifications. The manufacturing tolerances are based on DIN EN10034 for I- and H-girders [75], DIN EN10279 for U-beams [76], DIN EN10219-2 for cold-rolled hollow sections [77], and DIN EN10210-2 for hot-formed hollow sections [78].

The OWL-based extension of the DSTV-NC standard enables the inclusion and linking of production tolerance standards. This aligns the

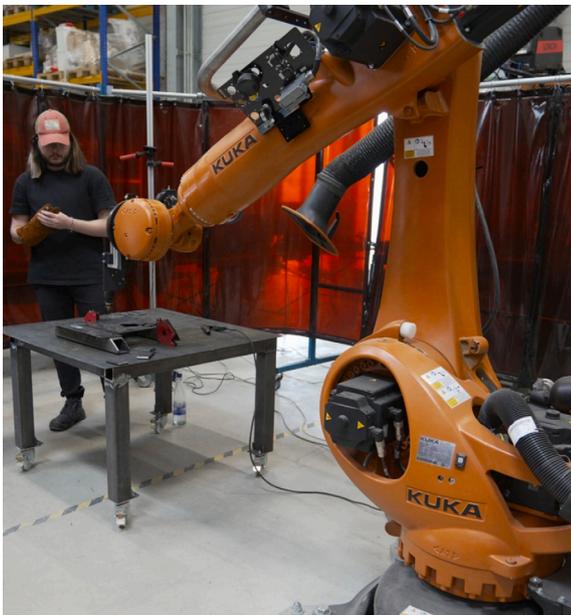


Fig. 6. A KUKA KR240 R2700 Robot in the plasma-cutting workspace.



Fig. 7. Custom torch for Hypertherm Powermax plasma-arc cutting system.

presented ontology with other ontologies that deal with the machine-readable description of these standards in the form of a knowledge base. The DSTV ontology includes only minimum and maximum bounds to ensure simplified process logic.

4.7. Properties of the ontology

This article presents the second iteration of an OWL-based extension to the DSTV-NC standard. It covers two of the seventeen process types described in the standard. The ontology currently consists of 68 classes, 30 object properties, which are relationships between classes, and 3 data type properties, which are links from classes to data types and literals.

5. Case study: Semantic DT for plasma-cutting

A case study of a plasma-cutting process sequence was chosen to further develop and evaluate the ontology. It aims to cut three hexagonal shapes from an IPE 220 steel beam. This can be achieved by cutting out parts of the web of the steel beam that are not needed for structural stability. The case study's objective is to realise an accessible, semantic DT by utilising the DSTV ontology concepts and the described extensions. This requires establishing a bidirectional system that connects the digital model of the machine and process to its real-world counterpart, enabling online robot control with low latency data and process feedback. By modelling all relevant parameters in a Linked Data fashion, capturing machine feedback, and later adding measurement data gathered with a specialised LiDAR-based measuring device, we aim to check tolerances and evaluate if process knowledge can be deduced from the gathered data.

5.1. Hardware and software setup

As shown in Fig. 6, a KUKA KR240 R2700 robot is used for plasma-cutting. It has a custom torch connected to a Hypertherm Powermax plasma arc cutting and gouging system (see Fig. 7). Both the robot and torch can be independently controlled using the KUKA|crc (Cloud Remote Control - CRC) framework as presented by Kerber et al. [79]. The control loop is executed on a Windows-based laptop with a six-core Intel i7 processor clocked at 2.60–4.50 GHz and 16 GB of DDR4

RAM (Double Data Rate 4 Random-access memory). The triple store is installed in a Docker environment on a low-power Intel NUC with a four-core Intel i5 processor clocked at 2.3 GHz. All devices are connected using a standard 5 GHz customer-grade Wireless Local Area Network (WLAN).

Setting up the software is straightforward. A Python-written control loop runs on the laptop, connected to the triple store via common Hypertext Transfer Protocol (HTTP) and the KUKA|crc main control unit (MCU) via Message Queuing Telemetry Transport (MQTT) [80], a lightweight, publish-subscribe, machine-to-machine network protocol designed for message queuing services. KUKA|crc communicates using interface-specific JavaScript Object Notation (JSON) formatted messages to and from the robot and the independently controlled torch. Both commands and data feedback are synchronised with the triple store based on a dockerised, open-source Blazegraph database [81]. An overview of the setup is shown in Fig. 9.

5.2. Initial modelling and data conversion

The first step in the case study process is modelling the necessary data. This includes information about the workpiece and desired cutouts, the machine and its metadata, and the process steps that will be executed.

The IPE220 workpiece, which includes three hexagonal cutouts, is modelled using Autodesk Advance Steel, a software for steel detailing [82]. The model is then exported as Industry Foundation Classes (IFC) Step Physical File (SPF) - version 4.0. To convert the resulting file, the IFCtoLBD.¹ converter is used [27,83] as described by M. Bonduel 2018, and Jyrki Oraskari 2023. For most BIM use cases, having the LBD data is sufficient. However, in this case, using the converter's functionality to include the ifcOWL description, which is linked via the *owl:sameAs* object property, is helpful as ifcOWL contains the full semantic richness of the original geometry in IFC SPF.

To create the DSTV data, the standard Linked Data query language, SPARQL [84], is then used to query the ifcOWL instances. Using the origin XY plane as a reference plane, instances of the corresponding DSTV classes can be inserted into the setup graph database.

¹ <https://github.com/jyrkioraskari/IFCtoLBD>.

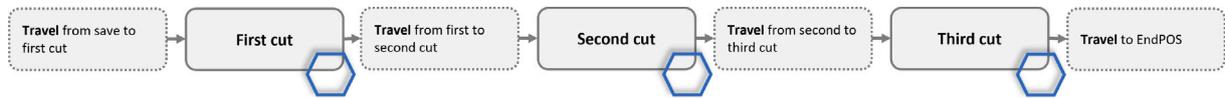


Fig. 8. Process sequence for the case study.

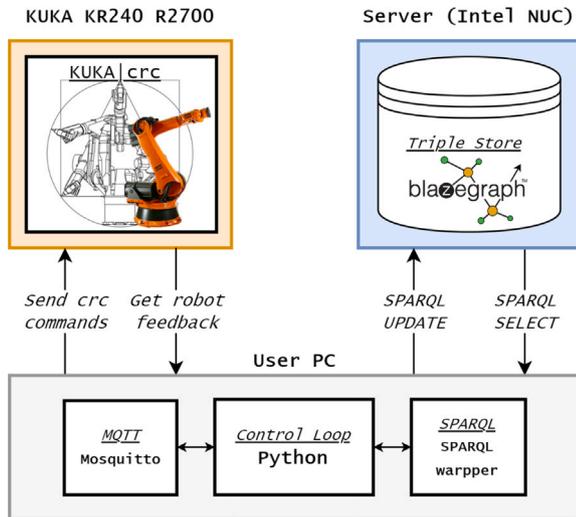


Fig. 9. Software setup for executing the plasma-cutting process.

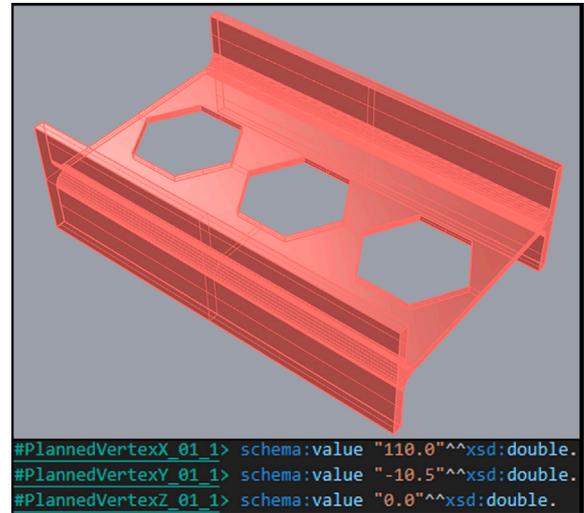


Fig. 10. Modelled IPE220 beam with three hexagonal cutouts including an excerpt of the resulting DSTV data for a vertex of an inner contour.

It is important to note that queries handling ifcOWL can be complex and depend on how elements in the IFC are created. For example, the data structure in the IFC of visually identical geometry can differ significantly depending on whether the geometry was created by sweeping, extrusion, or surface modelling. As a result, the queries used to retrieve the data vary. To address this challenge in the future, it should be tackled either within the IFC schema, by using a different exporter, or by directly creating the triples in the authoring software and bypassing IFC altogether.

Fig. 10 displays the beam and an excerpt of the DSTV triples generated.

The process sequence is generated in the next step by creating and linking instances of *ioc:Process*. This requires knowledge of the typical manufacturing steps of the machine being used and the desired level of detail. As shown in Fig. 8, the robotic process consists of three cutting processes (marked with a hexagon) and the travelling procedures from the robot's saved position to the first cut, between cuts, and to the end position (*EndPOS*) of the robot. The preferred level of detail is one process per hexagon cut. If metadata, such as timestamps, is required for each curve section of the hexagon cut, it is advisable to model them as individual sequential processes. To link the sequence to the workpiece information, the DSTV data is queried via SPARQL SELECT, the information is connected to the process instances as shown in Fig. 4, and the triples are pushed to the triple store via SPARQL UPDATE. To complete this step, additional process metadata is manually added. This includes resource description, such as the MQTT topic, workspace description, process constraints, and responsible person (*ioc:Actor*). Instances of *ioc>Status* are generated automatically, and all process metadata is synchronised with the server.

Finally, the robot operations are generated in Rhinoceros 3D using Grasshopper. The KUKA|crc plugin is utilised for path planning and generating robot commands for this application. A custom Python component in Grasshopper (GhPython) is employed to supply the required geometric information. This component queries the geometry of the feature values associated with the modelled *dstv:InternalContouring* via HTTP using the Python package RDFlib [85], and converts them into

Rhino curves. Within the Grasshopper workflow, only the translation of the workpiece in real-world coordinates (via the robot frame) and the robot model need to be manually set. Future improvements could include synchronising the KUKA|crc robot descriptions with the resource types used in the graph database.

Listing 1: KUKA|crc command for linear movement (MoveLin) in RDF.

```
1 @prefix inst: <http://baufest.org/dstv-test#> .
2 @prefix crc: <http://w3id.org/crc2#> .
3
4 inst:MoveLinCMD_091f7822-e356-5b9a a crc:MoveLinCMD
5 ;
6   crc:id "5edc7f61" ;
7   crc:cmd "MoveLin" ;
8   crc:dev "optimus" ;
9   crc:sync false ;
10  crc:tcpPose inst:tcpPose_1c5507bf-c9fc-5bf9 ;
11  crc:meta inst:meta_1c5507bf-c9fc-5bf9 .
12
13 inst:tcpPose_1c5507bf-c9fc-5bf9 crc:x "1952.71" ;
14   crc:y "118.69" ;
15   crc:z "708.65" ;
16   crc:a "0.0" ;
17   crc:b "90.0" ;
18   crc:c "0.0" .
19
20 inst:meta_1c5507bf-c9fc-5bf9 crc:speed "5.0" ;
21   crc:redAxis "0.0" ;
22   crc:softness "0.0" ;
23   crc:damping "100.0" ;
24   crc:atype "3" ;
25   crc:cvel "100.0" .
```

Next, the commands are converted to RDF triples with the help of another GhPython component. This is realised with a simple mapping

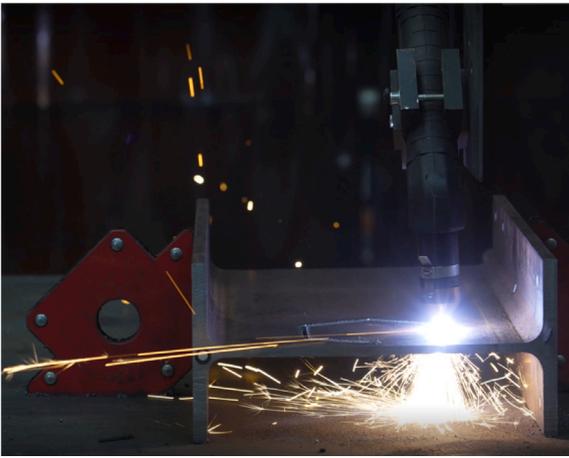


Fig. 11. Running plasma-cutting process of first hexagon cutout.

to a drafted *crc* ontology which duplicates the KUKA|*crc* JSON schema by creating concepts, as well as object and datatype properties arranged in RDF lists. While exemplarily, it would also be possible to convert the JSON externally with the RDF Mapping Language (RML) [86], using the GHPython in Grasshopper enables to dynamically create and preview the triples. Instead of immediately sending the KUKA|*crc* commands to execute the robotic process, the triples are uploaded to the triple store via SPARQL. This enables planning, reviewing and storing the commands for just-in-time execution. The commands can even be reused, as individual process parameters can be adapted just before the command is sent to the machine to execute via overwriting instances in the graph. An example of an instantiated Cloud Remote Control process in RDF is depicted in 1. In line 4, an instance of *crc:MoveLinCMD* for the device “optimus” (id of the robot) is created. It is linked to a *tcpPose* (line 12), describing the frame the robot will move to and the metadata of the command (line 19), which contains information like the execution speed of damping. Note that the IRIs have been shortened for better readability.

5.3. Process execution

The execution of the process is initiated by starting the process loop, which in the first step checks if there are any processes for the machine that has an *ioc:Status* that sets *ioc:isReady* to “true”. Initially, this is not the case, so the loop will check periodically until the status in the database is toggled externally, which, in this case, is done by a simple batch file. The above query now finds a process marked ready and continues to query for its data. 2 shows this subsequent query. In line 9 of this query, the process IRI that was retrieved previously is bound as a variable to query for the current values linked via *ioc:Method*.

Note that it is checked that the method is an *ioc:CurrentProcessDataState*, which ensures that a current version of the values is retrieved. Via the *ioc:Method* class, the KUKA|*crc* commands are linked as a list. Line 16 shows that this list is searched with the help of a SPARQL 1.1 Property Path, which can be recognised by the asterisk at the *list:hasNext**. This checks for a path of zero or more occurrences of this object property. Iterating through the list returns the *?Content* variable which is then used in OPTIONAL clauses to retrieve the robot command data independent of the command type that might not be known a priori. The requested robot commands are converted back from RDF to the KUKA|*crc* JSON structure and then sent to the MCU. At the same time, a new process state is created and inserted into the graph database.

Listing 2: SPARQL query to retrieve robot commands and metadata for a specific *ioc:Process*.

```

1 PREFIX ioc: <http://w3id.org/ioc#>
2 PREFIX list: <https://w3id.org/list#>
3 PREFIX crc: <http://w3id.org/crc#>
4
5 SELECT ?id ?cmd ?dev ?sync ?state ?redAxis ?softness
6 ?damping ?speed ?atype ?cvel ?x ?ty ?tz ?ta ?tb ?tc
7 ?x ?y ?z ?a ?b ?c WHERE {
8
9 BIND({<process>} AS ?process)
10 ?process a ioc:Process;
11     ioc:hasMethod ?_Method.
12 ?_Method a ioc:CurrentProcessDataState;
13     ioc:hasMethodValue ?Method.
14 ?Method a ioc:Method;
15     crc:hasCRCcmd ?cmdList.
16 ?cmdList list:hasNext* ?Node.
17 ?Node list:hasContents ?Content .
18
19 OPTIONAL{?Content crc:id ?id;
20     crc:cmd ?cmd;
21     crc:dev ?dev;
22     crc:sync ?sync.}
23 OPTIONAL{?Content crc:meta ?meta.
24     ?meta crc:atype ?atype;
25     crc:cvel ?cvel.}
26 OPTIONAL{?Content crc:jointPose ?jointPose.
27     ?jointPose crc:x ?x;
28     crc:y ?y;
29     crc:z ?z;
30     crc:a ?a;
31     crc:b ?b;
32     crc:c ?c.}
33 OPTIONAL{?Content crc:tcpPose ?tcpPose.
34     ?tcpPose crc:x ?x;
35     crc:y ?y;
36     crc:z ?z;
37     crc:a ?a;
38     crc:b ?b;
39     crc:c ?c.}
40 OPTIONAL{?Content crc:state ?state.}
41 OPTIONAL{?Content crc:meta ?meta.
42     ?meta crc:speed ?speed.}
43 OPTIONAL{?Content crc:meta ?meta.
44     ?meta crc:redAxis ?redAxis;
45     crc:softness ?softness;
46     crc:damping ?damping.}
47 }

```

Here, *ioc:isStarted* is set to true, the state is set as the current state, and a timestamp is generated and associated. As mentioned earlier, a modelled process can contain a number of commands depending on the level of detail required. In the case of a hexagonal cut, the process consists of 27 sequential commands. This includes turning the plasma on and off, as well as linear motion along the hexagon contour, which is divided into 25 commands for robotic procedural reasons.

Since the current logic does not cache commands, a single process for each command and its retrieval and transmission would lead to bad results since the time needed to retrieve and execute the commands from the triple store is about 25 ms. A limitation of this grouping is that metadata can only be dynamically adapted to the process for each hexagonal cut, not for each individual contour segment.

As the robot executes commands, the MCU periodically sends data feedback about the robot’s status. In this application, the “state” feedback determines if the robot is still busy with its task. If two consecutive “idle” states are retrieved, the process is marked as completed by inserting a new current state. According to the constraints (in this case,

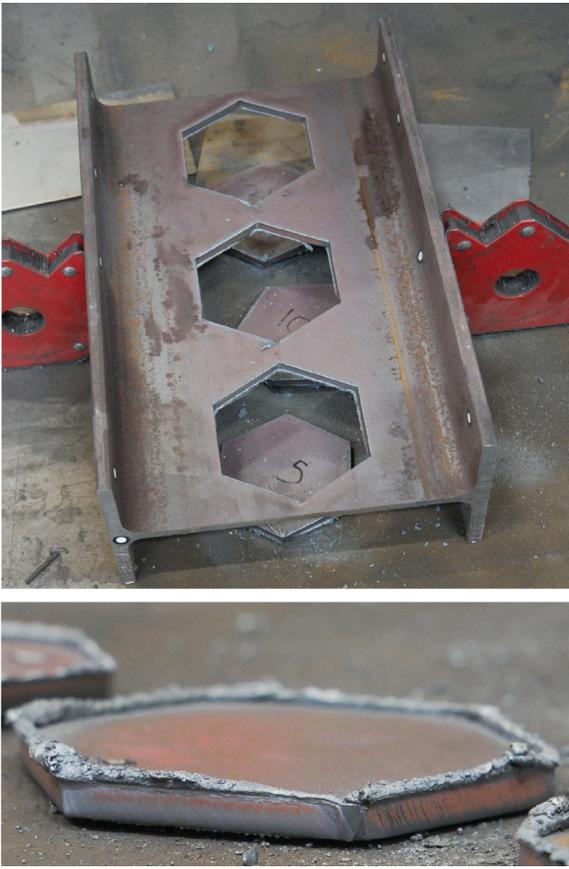


Fig. 12. Resulting beam with hexagon cutouts.

the previous process state must be set to finished), the next process state is set to ready. This is again retrieved by the control loop, which switches to the next process step. Fig. 11 shows the plasma torch cutting the first hexagonal cut.

After executing the final command that brings the robot to its end position, no additional processes are linked with *ioc:hasSuccessor*, which terminates the control loop. With all hexagonal cuts, the resulting beam is shown in Fig. 12. The detail of the cutout shape shows excess material around the edges.

5.4. Quality control

After plasma-cutting, the workpiece is scanned for quality control. Fig. 13 shows the scanning process using a handheld laser scanning device and its output. The output on the bottom shows an overlay of planned geometry and a scanned point cloud. Grey represents low deviation while yellow to red shows substantial deviation. Deviation analysis allows a detailed examination of deviations between planned and realised process steps in steel processing. The software can be extended with Python scripts to systematically map the obtained data. The vertices are detected, mapped to the *dstv:PlannedVertex* values and modelled as *dstv:MeasuredVertex* values to ensure comprehensive documentation. The three cutouts' visual results indicate varying cleanliness and precision along the cutting path. This is due to the use of different plasma-cutting parameters during the processes. The deviation increases from the first hexagon on the left to the last on the right. This is because each plasma cut's machining speed was set to 5, 10, and 20 mm per second.

Listing 3: SPARQL query to retrieve robot commands and metadata for a specific *ioc:Process*.

```

1 PREFIX ioc: <http://w3id.org/ioc#>
2 PREFIX prov: <http://http://www.w3.org/ns/prov#>
3 PREFIX crc: <http://w3id.org/crc#>
4 PREFIX list: <https://w3id.org/list#>
5 PREFIX dstv: <http://w3id.org/dstv#>
6
7 SELECT ?Name ?Class ?Feature (Count(?CRCcc)as ?count
8 )
9 ?id ?cmd ?dev ?speed ?Start ?Finish ?Diff WHERE{
10 ?Process a ioc:Process;
11   ioc:hasStatus ?_Status1;
12   ioc:hasStatus ?_Status2;
13   rdfs:label ?Name.
14 OPTIONAL{?Process ioc:hasStatus ?_Status1;
15           rdfs:type ?Class.
16 FILTER(strStarts(STR(?Class), "http://w3id.org/dstv
17 #"))}
18 OPTIONAL{?Process dstv:hasFeatureValue ?Feature.}
19 ?_Status1 ioc:hasStatusValue ?StatusStart;
20   prov:generatedAtTime ?TimeStart.
21   ?StatusStart ioc:isStarted "True"^^xsd:boolean;
22   ioc:isFinished "False"^^xsd:boolean.
23 ?_Status2 ioc:hasStatusValue ?StatusFinish;
24   prov:generatedAtTime ?TimeFinish.
25   ?StatusFinish ioc:isStarted "True"^^xsd:boolean;
26   ioc:isFinished "True"^^xsd:boolean.
27 BIND(xsd:dateTime(?Start)-xsd:dateTime(?Finish)as ?
28   Diff)
29 ?Process ioc:hasMethod ?_Method.
30 ?_Method ioc:hasMethodValue ?Method.
31 ?Method crc:hasCRCcmd ?CRC.
32 ?CRC list:hasNext ?CRC2;
33   list:hasNext* ?CRCcc.
34 ?CRC list:hasContents ?cont.
35 ?cont crc:id ?id;
36   crc:cmd ?cmd;
37   crc:dev ?dev;
38   crc:meta ?meta;
39   ?meta crc:speed ?speed.
40 }
41 GROUP BY ?Name ?Class ?Feature ?cmdCount ?id ?cmd ?dev
42 ?speed ?Start ?Finish ?Diff
43 ORDER BY ASC(?Start)

```

Table 3 presents the querying results described in 3, grouped as data contained in the DSTV, the linked ifcOWL instances, the machine data, and the process status feedback. In relation to the described execution of the three hexagons, the table provides a holistic view of the data that is linked to the processes. Based on the results of the analysis of the scanned, processed beam and the planned value for the feature, Table 4 displays the planned value, the measured values and the calculated deviations, for example, for a *dstv:Vertex* in millimetres. Following the illustration of the deviation of one vertex, Table 5 shows the averaged absolute deviations per hexagon for the deviation in the x, y and z directions. As the query does not include units for the different outputs, the data in the tables are not linked to units.

6. Evaluation

The following section focuses on the evaluation of the developed DSTV ontology and its extensions. The assessment carried out according to the methodology employed [63] involves verifying whether the Competency Questions (CQs) can be addressed unambiguously by querying

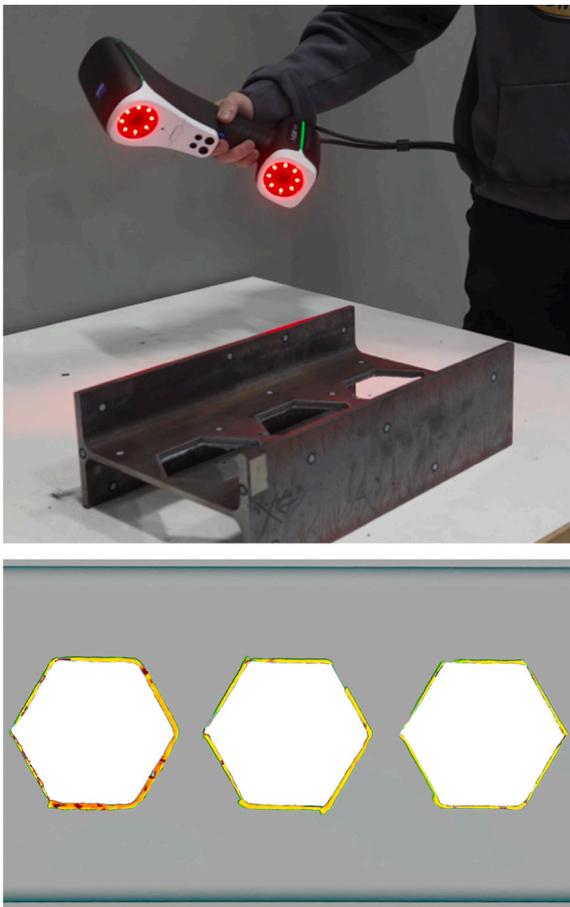


Fig. 13. Manual scan for quality control with the help of a handheld LiDAR device.

exemplary graph data. In this case, the data is not artificially modelled but generated in the case study described in detail in Section 5. Three key questions from Table 1 in relation to the findings of Tables 3–5 are selected.

CQ2: What are the measured values? As can be seen from the structure of the use case described, the workpiece was scanned after the plasma cutting. The data exported from the scanning-postprocessing software is automatically matched and linked to the data model. It can be queried (refer to Table 4, *Measured VY, VZ, VX*) and directly compared to planned values used in the manufacturing steps. The query that unambiguously retrieves the values can be used to adapt the workpiece's actual geometry data and for quality control after plasma cutting, extending the initial capabilities of DSTV-NC.

CQ6: What machines and what tools are used? For internal resource planning and to ensure the correct production process for machining, it is necessary to know and be able to answer at any time which machine is currently being used or has been used to produce a workpiece. The DSTV ontology enables the querying of machine data, both as generic *ioc:Resource* and a more detailed, specific robotic device (see *optimus*, the name of the robot) in Table 3 including all metadata used for the process. In addition, linked *start* and *finish* times for the robot movements for the cut-outs support optimal planning of logistics as well as process analysis for further optimisation.

CQ8: Can process knowledge be deduced from the resulting data? The Linked Data approach presented enables querying data from various processes, sensors, and machines, facilitating process analysis and optimisation. Table 5 displays the average absolute deviations for the three cutouts (i.e. Avg. (AbsDevX)). Furthermore, the robot's speed for the

different cuts (5.0/10.0/20 [mm/s]) is linked to the deviations. The table values show deviations increase at lower robot speeds than higher speeds for the presented plasma-cutting process. It can be concluded that the robot's speed impacts the quality of the plasma-cutting process. This is a well-known fact among trained personnel who operate these machines, particularly experts in robotic plasma-cutting. Lower speeds result in higher workpiece temperatures, leading to increased excess material. However, this correlation can be deduced from the Linked Data, a major first step towards machine-understandable manufacturing constraints and predictive optimisation using a semantic DT.

It can be concluded that the developed DSTV ontology can answer the three selected CQs. Table 3 also showed that CQ1 and CQ7 can be answered, even though the evaluation did not focus on this. Possible methods to enable CQ3, CQ4, and CQ5, which focus on tolerances and adaption, need further development in how the tolerances are modelled in a simplified way and thus, adaption can be deduced. Adjustment and adaptation are highly complex topics that need further rule-based approaches like SHACL to work practically.

7. Discussion

The primary research objective of this paper was to develop an ontology to describe concepts in the domain of steel construction needed to realise semantic DTs for robotic steel prefabrication processes. To enhance the practical applicability of this approach, the methodology presented in this paper involved developing an ontology based on an existing standard and demonstrating its use through the analysis of a real-world case study.

7.1. Key findings

First and foremost, our case study shows that a semantic DT for robotic plasma-cutting can be realised by applying a Linked Data approach with the help of the developed ontology. The DT enables machine control, process feedback and process adaptation to perform all key production functions centrally. The approach enables the control to interact with the existing robotic setup in the same low-level manner that other factory setups running the KUKA|crc framework already do. This indicates a high potential for generalisation and reusability for a wide variety of robotic processes in steel prefabrication. The use of the Grasshopper environment for path planning, converting and updating triples also shows that widely adopted, low-entry modelling environments can be employed to create functional data for the semantic DT.

The developed ontology was embedded in existing ontologies to create a structure that enables the modelling and linking of all required data. In the case study, the availability of the underlying data was proven by SPARQL queries that focussed on answering chosen competency questions. The resulting formalised information about the production process, the DT and the context of the DT can be of value for use cases like Predictive and optimised machine maintenance [87] or many others. This aligns with the analysis of other researchers in the topic of semantic DTs [9–11].

The presented approach enables continuous improvement by providing a detailed, data-driven understanding of each step in the fabrication process. This allows DT data like machine parameters and measured deviations to be available for downstream prefabrication processes. This can be especially valuable for following research on robotic assembly strategies, as deviations can be accessed as shown in the case study (see Table 4). This deviation could be addressed by adjusting connecting parts with tolerances — automatically or manually — ensuring structural integrity and proper fittings. The presented system can offer the flexibility of online control as deemed necessary by [29,30] due to dynamically updatable parameters in the graph. In addition, based on the data that was made available, structural

Table 3
Results of the query described in Listing 3.

| DSTV Data | | | ifcOWL | Machine Data | | | | Status Feedback | | |
|---------------------------------|--------------------------|------------------------|-----------------------------|--------------|----------|---------|-------|----------------------------------|----------------------------------|------------|
| Name | Class | Feature | Output | count | cmd | dev | speed | Start | Finish | Diff |
| Travel from Save to first cut | – | – | – | 2 | MoveLin | optimus | –1.0 | 2023-03-28T15:43:25.499584+02:00 | 2023-03-28T15:43:55.393978+02:00 | PT29.894S |
| First cut | dstv:Internal Contouring | inst:Contour List_0101 | inst:IfcVoiding Feature_186 | 27 | MoveLin | optimus | 5.0 | 2023-03-28T15:43:55.770522+02:00 | 2023-03-28T15:45:03.253795+02:00 | PT1M7.483S |
| Travel from first to second cut | – | – | – | 3 | MoveLin | optimus | –1.0 | 2023-03-28T15:45:03.847113+02:00 | 2023-03-28T15:45:32.219640+02:00 | PT28.372S |
| Second cut | dstv:Internal Contouring | inst:Contour List_0201 | inst:IfcVoiding Feature_170 | 27 | MoveLin | optimus | 10.0 | 2023-03-28T15:45:32.572661+02:00 | 2023-03-28T15:46:18.222345+02:00 | PT45.650S |
| Travel from second to third cut | – | – | – | 3 | MoveLin | optimus | –1.0 | 2023-03-28T15:46:18.720389+02:00 | 2023-03-28T15:46:36.752713+02:00 | PT18.032S |
| Third cut | dstv:Internal Contouring | inst:Contour List_0301 | inst:IfcVoiding Feature_154 | 27 | MoveLin | optimus | 20.0 | 2023-03-28T15:46:37.193713+02:00 | 2023-03-28T15:47:00.235659+02:00 | PT23.042S |
| Travel to EndPOS | – | – | – | 2 | MoveAxis | optimus | –1.0 | 2023-03-28T15:47:00.643774+02:00 | 2023-03-28T15:47:10.531371+02:00 | PT9.888S |

Table 4
Calculated deviation for one *dstv:Vertex* example.

| Feature | Planned Values | | | Measured values | | | Deviation values | | | |
|------------------|----------------|-----------|-----------|-----------------|------------|------------|------------------|-------------|-------------|--------|
| | PlannedVX | PlannedVY | PlannedVZ | MeasuredVX | MeasuredVY | MeasuredVZ | VXDeviation | VYDeviation | VZDeviation | tcpZ |
| inst:Vertex_01_1 | 110.0 | –10.5 | 0.0 | 109.858 | –9.255 | 0.304 | –0.141 | 1.244 | 0.304 | 708.65 |

Table 5
Averaged absolute deviations for the three cutouts.

| Name | cmd | dev | speed | tcpPose | Avg(AbsDevX) | Avg(AbsDevY) | Avg(AbsDevZ) | AbsTolXYZ | tcpZ |
|------------|---------|---------|-------|---|--------------|--------------|--------------|-----------|--------|
| First cut | MoveLin | optimus | 5.0 | inst:tcpPose_8c328310-a47c-5ac7-be84-d3db1d2007c3 | 0.368 | 0.547 | 4.079 | ±0.6 | 708.65 |
| Second tut | MoveLin | optimus | 10.0 | inst:tcpPose_3050c77a-63b8-5869-a864-5635105b5023 | 0.231 | 0.504 | 3.399 | ±0.6 | 708.65 |
| Third tut | MoveLin | optimus | 20.0 | inst:tcpPose_706a2a54-ba7f-5750-b8ed-6d99f1802774 | 0.190 | 0.413 | 3.086 | ±0.6 | 708.65 |

engineers can receive assistance in selecting optimal design parameters, leading to more efficient and higher-quality production.

While most results met our expectations, the ability to infer process knowledge (see Table 5 and CQ8) surpassed them. The realised semantic DT is based on ontologies that are, by design, both machine- and human-readable [38,40]. In the case study, the Linked Data approach enabled us to cleanly infer knowledge about the influence of machine parameters on the deviations of the fabricated workpiece. This proved in a practical application that process dependencies and knowledge can be formalised using the semantic DT approach. We see huge potential for better knowledge transfer, enhancing collaboration, and promoting assistant systems, thus leading to more accurate and adaptable prefabrication processes.

7.2. Contributions

The approach presented in this article demonstrates advantages over the standard DSTV-NC method. A benefit is the measured as-built values. While it is possible to incorporate extensions into the standard, standardisation is often protracted. In contrast, the modular framework of Linked Data (LD)[88] and the utilisation of existing ontologies preclude the necessity of redesigning new definitions when existing ontologies exist.

Additionally, the dereferencing capability and unique naming intrinsic to LD principles [21] facilitate the establishment of a single source of truth. This ensures unambiguous data access, allowing information generated in the early stages of fabrication to be available during subsequent assembly phases whenever needed.

In addition to contributing an openly available ontology for the domain, this paper can demonstrate the feasibility of using Linked Data approaches and semantic DTs for full-scale production processes, thus making a significant contribution to the exploration of robotic processes in the steel construction domain. Developing and implementing an

ontology based on a practical case study adds complexity to the task. However, this allows us to demonstrate that the DSTV ontology is not just a theoretical construct but can contribute to addressing the domain challenges described throughout this article.

7.3. Limitations

This paper focused solely on two classes of the DSTV-NC production processes, leaving other use cases and potential challenges unexplored. The current approach lacks detailed descriptions for more complex workpieces. Moreover, the process cannot accommodate a wider variety of components without comprehensive part descriptions, limiting its versatility and applicability to more intricate or non-standard designs.

Although the paper successfully shows a prototype integration of the semantic DT, other existing robot control systems may have different specifications and configurations that limit support for semantic web interfaces or require extensive modifications. This also applies to the Grasshopper environment, which is not considered a production environment and may not be fully compatible with other production tools and software, potentially leading to issues of workflow integration and process efficiency. As described in the case study, the demonstrated quality control processes have not yet been automated. The prototypical solution and the software used provided a Python plug-in interface that allowed easy access to the measurements in the desired format. This may not be the case for machines with integrated sensors and measurement devices.

As previously explained, the availability of feedback and deviation data holds significant potential for assembly in steel construction. The current approach only stores information from part production and does not yet describe assembly processes.

Challenges persist in comparing indirect deviations, ensuring compatibility with existing steel software, and addressing issues with data loss and misaligned properties in IFC. Furthermore, the complexity of

the data input hindered the realisation of a DT with real-time simulation capabilities that exceed the robot's general kinematics, indicating areas for future research and development.

7.4. Future work

To improve the understanding of the limitations and constraints of the presented approach, future research should focus on developing concepts for all remaining types of DSTV-NC production processes by mapping them in OWL and RDF. Additionally, exploring the interconnection and alignment with existing model views in the steel construction domain is necessary to provide interoperability and maximise effectiveness and impact in the industry. More importantly, it is advisable to conduct research on how to model and link steel assembly data, as it is typically the subsequent step in the life cycle stage. Based on the WISCON interface and the resulting BFS-RL-03-108 [18], we will focus on processing and using part production information in assembly and describing deviations during assembly. This will enable more accurate, material-saving manufacturing. Furthermore, combined with defined tolerances for parts and assemblies, it will allow for the digital classification of quality manufacturing classes in steel construction. Suppose the Linked Data approach used in this paper is to be extended to later stages in the life cycle, aligning it with the immediately succeeding manufacturing process is important. To accomplish this, it is necessary to conduct additional empirical testing, validation, and conformity checks in various industrial settings.

8. Conclusion

This paper presented the development of an ontology that maps and extends the DSTV-NC standard interface to enable a semantic DT for robotic steel prefabrication. The relevant state of the art was summarised, and a structured guide to developing the concepts and relations of the ontology was introduced as a methodology. The creation of a semantic DT realising a robotic plasma-cutting process was used as a case study for a real-world robotic process to demonstrate the concept's applicability in practice. The workpiece, machines, devices, and robotic command metadata information were described using semantic web technologies. The DSTV-NC instructions, modelled as triples in a graph database, served as a linking element for all this information. During the execution of the process sequence, we utilised Linked Data concepts to realise the bidirectional nature of a DT. These concepts were synchronised with the interfaces of the KUKA|rc framework, allowing us to save and link process metadata, manipulate it dynamically, or store it for quality considerations in later life cycle stages of the element. After machining, quality control measures were performed by scanning the resulting workpiece. The gathered measurements were added and linked to the process model. The resulting database can be queried to check initial planning against measured deviations. Additionally, the data can be used to deduce process knowledge, demonstrating the potential of using data described according to the structure of the developed ontology to optimise future production procedures.

CRedit authorship contribution statement

Lukas Kirner: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Victoria Jung:** Writing – review & editing, Writing – original draft, Project administration, Investigation, Conceptualization. **Jyrki Oraskari:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sigrid Brell-Cokcan:** Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Full implementation details and datamodels are available on request.

Acknowledgements

This work is part of the research project *Bauplanungsorientiertes Fertigungsmanagement im Stahlbau 4.0* (BauFeSt 4.0 – project number 21690 N/FE1). It is carried out in cooperation with industrial partners and financially supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) through the German Federation of Industrial Research Associations (AiF). It includes content developed within the *EConoM* research project, funded by the Federal Ministry of Digital Affairs and Transport (BMDV) under the InnoNT initiative (grant number 19OI22009F). Additionally, work was also conducted as part of the project *Leitfaden für Informationsnetzwerke im Bauwesen* which is funded by Ministry of Regional Identity, Local Government, Building and Digitalisation of the State of North Rhine-Westphalia (MHKBD) (grant number 35.05.04.01-IB-2022-007).

References

- [1] P.F. Rocha, N.O. Ferreira, F. Pimenta, N.B. Pereira, Impacts of prefabrication in the building construction industry, *Encyclopedia* 3 (1) (2023) 28–45, <https://dx.doi.org/10.3390/encyclopedia3010003>, URL <https://www.mdpi.com/2673-8392/3/1/3>.
- [2] F. Sikstrom, M. Ericsson, P. Nylén, A.-K. Christiansson, 3D-scanning for weld distortion measuring, in: 2006 IEEE Instrumentation and Measurement Technology Conference Proceedings, 2006, pp. 2132–2137, <https://dx.doi.org/10.1109/IMTC.2006.328524>.
- [3] H. Wilmsmeyer, M. Flasskamp, M.-O. Schierenberg, Intelligente korrektur eines schweißroboters, in: J. Jasperneite, V. Lohweg (Eds.), *Kommunikation und Bildverarbeitung in der Automation*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, pp. 283–294, https://dx.doi.org/10.1007/978-3-662-55232-2_22.
- [4] DIN e.V. (Hrsg.), *DIN EN 1090-2:2018-09, Execution of Steel Structures and Aluminium Structures - Part 2: Technical Requirements for Steel Structures*, Beuth Verlag GmbH, Berlin, <https://dx.doi.org/10.31030/2758614>.
- [5] H. Wilmsmeyer, M. Flasskamp, T. Jungeblut, M.-O. Schierenberg, *Automatisierung, Industrieroboter, MontageIntelligenter schweißroboter: Selbständige reaktion eines schweißroboters auf große toleranzen*, *Werkstattstechnik* (2016) 21.
- [6] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, Y. Xia, Survey of robot 3D path planning algorithms, *J. Control Sci. Eng.* 2016 (1) (2016) 7426913, <https://dx.doi.org/10.1155/2016/7426913>.
- [7] MFG Automation Systems, *Understanding adaptive robotic path planning in manufacturing*, 2024, URL <https://www.mfgautosys.com/understanding-adaptive-robotic-path-planning-in-manufacturing/>. (Accessed 23 June 2024).
- [8] H.-C. Song, J.-B. Song, Precision robotic deburring based on force control for arbitrarily shaped workpiece using CAD model matching, *Int. J. Precis. Eng. Manuf.* 14 (1) (2013) 85–91, <https://dx.doi.org/10.1007/s12541-013-0013-2>.
- [9] S.R. Bader, M. Maleshkova, The semantic asset administration shell, in: M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, Y. Sure-Vetter (Eds.), *Semantic Systems. the Power of AI and Knowledge Graphs*, Springer International Publishing, Cham, 2019, pp. 159–174, https://dx.doi.org/10.1007/978-3-030-33220-4_12.
- [10] J.M. Gómez-Berbis, A. de Amescua-Seco, SEDIT: Semantic digital twin based on industrial IoT data management and knowledge graphs, in: R. Valencia-García, G. Alcaraz-Mármol, J. Del Cioppo-Morstadt, N. Vera-Lucio, M. Bucaram-Leverone (Eds.), *Technologies and Innovation*, Springer International Publishing, Cham, 2019, pp. 178–188, https://dx.doi.org/10.1007/978-3-030-34989-9_14.
- [11] C. Boje, A. Guerriero, S. Kubicki, Y. Rezgüi, Towards a semantic construction digital twin: Directions for future research, *Autom. Constr.* 114 (2020) 103179, <https://dx.doi.org/10.1016/j.autcon.2020.103179>, URL <https://www.sciencedirect.com/science/article/pii/S0926580519314785>.
- [12] N. Shadbolt, T. Berners-Lee, W. Hall, The semantic web revisited, *IEEE Intell. Syst.* 21 (3) (2006) 96–101, <https://dx.doi.org/10.1109/MIS.2006.62>.

- [13] bauforumstahl e.V., Standardbeschreibung von stahlbauteilen für die NC-Steuerung (ASCII), 2003, URL https://bauforumstahl.de/wp-content/uploads/2024/04/109_Standartbeschreibung_von_Stahlbau-Teilen_8.0.pdf. (Accessed 19 June 2024).
- [14] bauforumstahl e.V., Standardbeschreibung von stahlbauteilen für die NC-Steuerung (XNC): BFS-RL 03-105, 2006, URL https://bauforumstahl.de/wp-content/uploads/2024/04/105_nc-xml.pdf. (Accessed 19 June 2024).
- [15] Jonas Hippe, *Entwicklung Einer Prototypischen Schnittstelle Zur Ansteuerung von Industrierobotern Über den DSTV-NC (Master's thesis)*, RWTH Aachen University, Aachen, 2021.
- [16] Robert McNeel & Associates, Grasshopper: Algorithmic Modeling for Rhino, 2024, URL <https://www.grasshopper3d.com>. Online; (Accessed 26 June 2024).
- [17] bauforumstahl e.V., Bauforumstahl news: Stahlbau 2/2018, Stahlbau 87 (2) (2018) 173–190, <http://dx.doi.org/10.1002/stab.201870214>.
- [18] bauforumstahl e.V., Standartbeschreibung von strukturen in der stahlbaufertigung (anbauschchnittstelle): BFS-RL 03-108, 2013, URL https://dstv.deutscherstahlbau.de/fileadmin/user_upload/bauforumstahl.de/wissen/normen-und-richtlinien/bfs-richtlinien/108_Anbauschchnittstelle-2013-10-24.pdf. (Accessed 19 June 2024).
- [19] W.W.W. Consortium, et al., Best practices for publishing linked data, 2014, URL <https://www.w3.org/TR/ld-bp/>. [Online; Accessed 24 July 2021], World Wide Web Consortium.
- [20] E. Hyvönen, J. Tuominen, M. Alonen, E. Mäkelä, Linked data Finland: A 7-star model and platform for publishing and re-using linked datasets, in: V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, A. Tordai (Eds.), *The Semantic Web: ESWC 2014 Satellite Events*, Springer International Publishing, Cham, 2014, pp. 226–230, http://dx.doi.org/10.1007/978-3-319-11955-7_24.
- [21] T. Berners-Lee, Linked data, 2006, URL <https://www.w3.org/DesignIssues/LinkedData>. Online; (Accessed 15 October 2023).
- [22] V. Jung, K.-H. Häfele, A. Geiger, M. Huhn, S. Stiegeler, S. Brell-Cokcan, Enabling interoperability in steel construction – new concepts for using IFC for steel fabrication, in: NSCC Sweden, NORDIC STEEL Construction Conference 2024 (2024) Luleå, Schweden, 26.06.2024–28.06.2024, 2024, <http://dx.doi.org/10.5281/zenodo.12509159>.
- [23] The International Organization for Standardization, Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries –Part 1: Data Schema, ISO, Geneva, Switzerland, 2024, URL <https://www.iso.org/standard/84123.html>. (Accessed 9 August 2024).
- [24] L.F. Rocha, P. Tavares, P. Malaca, C. Costa, J. Silva, G. Veiga, Beam for the steel fabrication industry robotic systems, in: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Vol. 34, IAARC Publications, 2017, URL <https://www.iaarc.org/publications/fulltext/ISARC2017-Paper089.pdf>. (Accessed 09 August 2024).
- [25] DIN e.V. (Hrsg.), Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries - Part 1: Data Schema (ISO 16739-1:2018); English Version EN ISO 16739-1:2020, Beuth Verlag GmbH, Berlin, <http://dx.doi.org/10.31030/3144077>.
- [26] J. Beetz, J. van Leeuwen, d. Vries, IFCOWL: A case of transforming EXPRESS schemas into ontologies, AI EDAM 23 (2009) 89–101, <http://dx.doi.org/10.1017/S0890060409000122>.
- [27] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, R. Klein, The IFC to linked building data converter - current status, 2018, URL <https://www.semanticscholar.org/paper/The-IFC-to-linked-building-data-converter-current-Bonduel-Oraskari/945040b77455d09b3f37f975fb18ab12624982d6>.
- [28] D.I. Ruby, *Steel Construction Process Study*, Chicago, 2004, URL <https://www.aisc.org/technical-resources/research/researchlibrary/steel-construction-process-study-tolerances/>.
- [29] G. Hoffmeier, Konzept einer Fehlerkorrektur der Laserbasierten Bauteil-lokalisierung für Schweißrobotersysteme, in: *Berichte aus der Robotik*, Shaker Verlag, Aachen, 2017, <http://dx.doi.org/10.2370/9783844055818>.
- [30] G. Bolmsjö, M. Olsson, P. Cederberg, Robotic arc welding – trends and developments for higher autonomy, Ind. Robot 29 (2) (2002) 98–104, <http://dx.doi.org/10.1108/01439910210419088>.
- [31] D. Ni, A.W.W. Yew, S.K. Ong, A.Y.C. Nee, Haptic and visual augmented reality interface for programming welding robots, Adv. Manuf. 5 (3) (2017) 191–198, <http://dx.doi.org/10.1007/s40436-017-0184-7>.
- [32] Dr.-Ing. Rolf Hedderich, *Selbstlernende schweißroboter: Innovationsprojekt: Intelligente verarbeitung von großbauteilen mit großen toleranzen*, 2017, <https://www.its-owl.de/projekte/abgeschlosseneprojekte/innovationsprojekte/details/intelligente-verarbeitung-von-grossbauteilen-mit-grossen-toleranzen/>. (Online; Accessed 07 August 2020).
- [33] Fakultät Maschinenbau: Institut für Produktionssysteme, ROPROF: Roboterschweißen von individuell gestalteten stahlbauprofilen in kleinsten stückzahlen, 2024, URL <https://ips.mb.tu-dortmund.de/forschen-beraten/forschungsprojekte/abgeschlossene-projekte/roprof/>. Accessed 10 April 2024.
- [34] R. Fischer, *Prozess- und modellorientierung des stahlbaus: Ein beitrag aus der praxis*, in: C. Motzko (Ed.), *Praxis des Bauprozessmanagements*, Ernst & Sohn, Berlin, 2013, pp. 193–212, <http://dx.doi.org/10.1002/9783433602973.ch8>.
- [35] M. Sabou, S. Biffi, A. Einfalt, L. Krammer, W. Kastner, F.J. Ekaputra, Semantics for cyber-physical systems: A cross-domain perspective, *Semantic Web* 11 (2020) 115–124, Available <https://api.semanticscholar.org/CorpusID:204783136>. (Accessed 10 August 2023).
- [36] A. Wong, P. Ray, N. Parameswaran, J. Strassner, Ontology mapping for the interoperability problem in network management, *IEEE J. Sel. Areas Commun.* 23 (10) (2005) 2058–2068, <http://dx.doi.org/10.1109/JSAC.2005.854130>.
- [37] A. Gyrard, A. Zimmermann, A. Sheth, Building IoT-based applications for smart cities: How can ontology catalogs help? *IEEE Internet Things J.* 5 (5) (2018) 3978–3990, <http://dx.doi.org/10.1109/JIOT.2018.2854278>.
- [38] Q. Cao, F. Giustozzi, C. Zanni-Merk, F. de Bertrand de Beuvron, C. Reich, Smart condition monitoring for industry 4.0 manufacturing processes: An ontology-based approach, *Cybern. Syst.* 50 (2) (2019) 82–96, <http://dx.doi.org/10.1080/01969722.2019.1565118>.
- [39] T.R. Gruber, A translation approach to portable ontology specifications, *Knowl. Acquis.* 5 (2) (1993) 199–220, <http://dx.doi.org/10.1006/knac.1993.1008>, URL <https://www.sciencedirect.com/science/article/pii/S1042814383710083>.
- [40] L. Patil, D. Dutta, R. Sriram, Ontology-based exchange of product data semantics, *IEEE Trans. Autom. Sci. Eng.* 2 (3) (2005) 213–225, <http://dx.doi.org/10.1109/TASE.2005.849087>.
- [41] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, *Autom. Constr.* 63 (2016) 100–133, <http://dx.doi.org/10.1016/j.autcon.2015.12.003>, URL <https://www.sciencedirect.com/science/article/pii/S0926580515002435>.
- [42] S. Törmä, Semantic linking of building information models, in: 2013 IEEE Seventh International Conference on Semantic Computing, 2013, pp. 412–419, <http://dx.doi.org/10.1109/ICSC.2013.80>.
- [43] Y. Zheng, S. Törmä, O. Seppänen, A shared ontology suite for digital construction workflow, *Autom. Constr.* 132 (2021) 103930, <http://dx.doi.org/10.1016/j.autcon.2021.103930>, URL <https://www.sciencedirect.com/science/article/pii/S0926580521003812>.
- [44] L. Kirner, P.R. Wildemann, S. Brell-Cokcan, ioc:process – ein neuer ansatz einer bauprozess-ontologie für die unternehmensübergreifende kollaboration, in: S. Brell-Cokcan, R.H. Schmitt (Eds.), *IoC - Internet of Construction : Informationsnetzwerke Zur Unternehmensübergreifenden Kollaboration in Den Fertigungsketten Des Bauwesens*, Springer Fachmedien Wiesbaden, Wiesbaden, 2024, pp. 229–278, http://dx.doi.org/10.1007/978-3-658-42544-9_9.
- [45] L. Kirner, J. Oraskari, P.R. Wildemann, Internet of Construction Process Ontology (ioc), Chair of Individualized Production in Architecture, RWTH Aachen University, 2023, URL <http://w3id.org/ioc>. (Accessed 13 December 2023).
- [46] M. Bonduel, *A Framework for a Linked Data-based Heritage BIM (Ph.D. thesis)*, Faculty of Engineering Technology, KU Leuven, 2021, Available at <https://lirias.kuleuven.be/3416395&lang=en>. (Accessed 08 August 2024).
- [47] R. Cacciotti, J. Valach, P. Kuneš, M. Čerňanský, M. Blaško, P. Křemen, Monument damage information system (mondis): an ontological approach to cultural heritage documentation, *ISPRS Ann. Photogram. Remote Sens. Spat. Inf. Sci.* II-5/W1 (2013) 55–60, <http://dx.doi.org/10.5194/isprsannals-II-5-W1-55-2013>, URL <https://isprs-annals.copernicus.org/articles/II-5-W1/55/2013/>.
- [48] E.V. Kalemi, F. Cheung, A.-R.H. Tawil, P. Patlakas, K. Alyania, ifcOWL-DFMA a new ontology for the offsite construction domain, in: LDAC 2020-Proceedings of the 8th Linked Data in Architecture and Construction Workshop, 2020, URL <https://ceur-ws.org/Vol-2636/08paper.pdf>. (Accessed 08 August 2024).
- [49] N.M. El-Gohary, T.E. El-Diraby, Domain ontology for processes in infrastructure and construction, *J. Construct. Eng. Manage.* 136 (7) (2010) 730–744, [http://dx.doi.org/10.1061/\(ASCE\)CO.1943-7862.0000178](http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000178).
- [50] S. Lee, S. Isaac, T. Bock, An ontology for process information modeling, in: M. Malaska, R. Heikkilä (Eds.), *Proceedings of the 32nd International Symposium on Automation and Robotics in Construction and Mining*, ISARC 2015, International Association for Automation and Robotics in Construction (IAARC), Oulu, Finland, 2015, pp. 1–7, <http://dx.doi.org/10.22260/ISARC2015/0032>.
- [51] J.L. Martinez Lastra, I.M. Delamer, Ontologies for production automation, in: T.S. Dillon, E. Chang, R. Meersman, K. Sycara (Eds.), *Advances in Web Semantics I: Ontologies, Web Services and Applied Semantic Web*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 276–289, http://dx.doi.org/10.1007/978-3-540-89784-2_11.
- [52] M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, K. Janowicz, BOT: The building topology ontology of the W3C linked building data group, *Semant. Web* 12 (1) (2020) 143–161, <http://dx.doi.org/10.3233/SW-200385>.
- [53] A. Wagner, W. Sprenger, C. Maurer, T.E. Kuhn, U. Rüppel, Building product ontology: Core ontology for linked building product data, *Autom. Construct.* 133 (2022) 103927, <http://dx.doi.org/10.1016/j.autcon.2021.103927>.
- [54] M. Grieves, *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*, first ed., Vol. 8, McGraw-Hill Education, New York, NY, 2006, p. 319.
- [55] M. Grieves, Origins of the digital twin concept, 2016, p. 7, <http://dx.doi.org/10.13140/RG.2.2.26367.61609>.
- [56] T.H.-J. Uhlemann, C. Lehmann, R. Steinhilper, The digital twin: Realizing the cyber-physical production system for industry 4.0, *Procedia CIRP* 61 (2017) 335–340, <http://dx.doi.org/10.1016/j.procir.2016.11.152>, URL <https://www.sciencedirect.com/science/article/pii/S2212827116313129>. The 24th CIRP Conference on Life Cycle Engineering.

- [57] N. Kousi, C. Gkourmelos, S. Aivaliotis, C. Giannoulis, G. Michalos, S. Makris, Digital twin for adaptation of robots' behavior in flexible robotic assembly lines, *Procedia Manuf.* 28 (2019) 121–126, <http://dx.doi.org/10.1016/j.promfg.2018.12.020>, URL <https://www.sciencedirect.com/science/article/pii/S2351978918313623>. 7th International conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2018).
- [58] M. Kümpel, C.A. Mueller, M. Beetz, Semantic digital twins for retail logistics, in: M. Freitag, H. Kotzab, N. Megow (Eds.), *Dynamics in Logistics: Twenty-Five Years of Interdisciplinary Logistics Research in Bremen, Germany*, Springer International Publishing, Cham, 2021, pp. 129–153, http://dx.doi.org/10.1007/978-3-030-88662-2_7.
- [59] D. Schraudner, A. Harth, A restful interaction model for semantic digital twins, in: R. García-Castro (Ed.), *SeDIT 2022, Third International Workshop on Semantic Digital Twins. Proceedings*, in: *CEUR Workshop Proceedings*, Vol. 3291, CEUR, 2022, URL <https://publica.fraunhofer.de/handle/publica/446006>. Online; (Accessed 26 June 2024).
- [60] F. Manola, E. Miller, B. McBride, et al., RDF primer, W3C Recomm. 10 (1–107) (2004) 6, URL <https://www.w3.org/TR/rdf-primer/>. (Accessed 08 June 2024).
- [61] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities, *Sci. Am.* (2001) 34–43, <http://dx.doi.org/10.1038/scientificamerican0501-34>.
- [62] D.L. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C Recomm. 10 (10) (2004) URL <https://www.w3.org/TR/owl-features/>. (Accessed 08 June 2024).
- [63] N.F. Noy, D.L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001, URL <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html>. (Accessed 08 June 2024).
- [64] M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, K. Janowicz, BOT: The building topology ontology of the W3C linked building data group, *Semant. Web* 12 (1) (2020) 143–161, <http://dx.doi.org/10.3233/SW-200385>.
- [65] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, *Autom. Constr.* 63 (2016) 100–133, <http://dx.doi.org/10.1016/j.autcon.2015.12.003>.
- [66] *Ontology for Construction Processes*, VDI-Gesellschaft Bauen und Gebäude-technik, 2023, URL <https://www.dinmedia.de/en/technical-rule/vdi-ee-2558/371821816>. (Accessed 08 August 2024). VDI-EE 2558:2023-11.
- [67] M. Rasmussen, M. Lefrançois, M. Bonduel, C. Hviid, J. Karlshøj, OPM: An ontology for describing properties that evolve over time, in: *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*, Vol. 2159, 2018, pp. 24–33, URL <https://backend.orbit.dtu.dk/ws/portalfiles/portal/152321814/Untitled.pdf>.
- [68] schema.org project, *Schema.org - schemas and supporting software*, 2024, URL <https://github.com/schemaorg/schemaorg>. (Accessed 11 April 2024).
- [69] L. Polo, D. Berrueta, MUO - measurement units ontology, 2008, URL https://akswnc7.informatik.uni-leipzig.de/dstreitmatter/archivo/elite.polito.it/ontologies--muo-vocab--owl/2020.06.10-182923/ontologies--muo-vocab--owl_type=generatedDocu.html. (Accessed 24 June 2024). Working draft.
- [70] SysML 1.2 Revision Task Force, *Quantities, units, dimensions, values (QUDV)*, 2009, URL <https://www.omg.org/spec/SysML/1.2/>. (Accessed 24 June 2024). Working draft.
- [71] H. Rijgersberg, M. van Assem, J.L. Top, *Ontology of units of measure and related concepts*, *Semant. Web J.* 4 (1) (2013) 3–13, <http://dx.doi.org/10.3233/SW-2012-0062>.
- [72] R. Hodgson, P.J. Keller, J. Hodges, J. Spivak, QUDT - Quantities, Units, Dimensions and Data Types Ontologies, Technical report, NASA, 2024, URL <https://www.qudt.org/>. (Accessed 08 August 2024).
- [73] M. Lefrançois, A. Zimmermann, The unified code for units of measure in RDF: ucum and other UCUM datatypes, in: *The Semantic Web: ESWC 2018 Satellite Events: ESWC 2018 Satellite Events*, Heraklion, Crete, Greece, June 3–7, 2018, *Revised Selected Papers* 15, Springer, 2018, pp. 196–201, http://dx.doi.org/10.1007/978-3-319-98192-5_24.
- [74] H. Knublauch, D. Kontokostas, *Shapes constraint language (SHAFL)*, 2017, URL <https://www.w3.org/TR/shacl/>. Online; (Accessed 15 October 2023).
- [75] DIN e.V. (Hrsg.), *DIN EN 10034:1994-03, Structural Steel I and H Sections; Tolerances on Shape and Dimensions; German Version EN 10034:1993*, Beuth Verlag GmbH, Berlin, <http://dx.doi.org/10.31030/2557002>.
- [76] DIN e.V. (Hrsg.), *DIN EN 10279:2000-03, Hot Rolled Steel Channels - Tolerances on Shape, Dimensions and Mass; German Version EN 10279:2000*, Beuth Verlag GmbH, Berlin, <http://dx.doi.org/10.31030/8031927>.
- [77] DIN e.V. (Hrsg.), *DIN EN 10219-2:2019-07, Cold Formed Welded Structural Hollow Sections of Non-Alloy and Fine Grain Steels - Part 1: Technical Delivery Conditions, German Version EN 10219-1:2006*, Beuth Verlag GmbH, Berlin, <http://dx.doi.org/10.31030/9666332>.
- [78] DIN e.V. (Hrsg.), *DIN EN 10210-2:2019-07, Hot Finished Steel Structural Hollow Sections - Part 2: Tolerances, Dimensions and Sectional Properties; German Version EN 10210-2:2019*, Beuth Verlag GmbH, Berlin, <http://dx.doi.org/10.31030/3058497>.
- [79] E. Kerber, J.L. Fahrendholz, S. Brell-Cokcan, P. Dewald, R. Sharma, U. Reisgen, Dynamic WAAM: adaptive processes for equivalent contact surface (ECS) optimization, *Constr. Robot.* 7 (3–4) (2023) 309–328, <http://dx.doi.org/10.1007/s41693-023-00113-7>.
- [80] OASIS Open, MQTT version 5.0, 2019, URL <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>. Online; (Accessed 15 April 2024).
- [81] Blazegraph, *Blazegraph DB*, 2024, URL <https://blazegraph.com/>. (Accessed 25 June 2024).
- [82] Autodesk Inc., *Autodesk advanced steel*, 2024, URL <https://www.autodesk.com/products/advance-steel/overview>. (Accessed 25 June 2024).
- [83] J. Oraskari, M. Bonduel, K. McGlenn, P. Pauwels, F. Priyatna, A. Wagner, V. Kukkonen, S. Steyskaland, J. Lehtonen, M. Lefrançois, IFctoLBD: IFctoLBD v 2.43.3, 2023, URL <https://github.com/jyrkioraskari/IFctoLBD>. Online; (Accessed 26 June 2024).
- [84] S. Harris, A. Seaborne, E. Prud'hommeaux, *SPARQL 1.1 Query Language, W3C Recommendation, W3C*, 2013, <https://www.w3.org/TR/sparql11-query/>, Online; (Accessed 15 October 2023).
- [85] R. Community, *RDFLib: A python library for working with RDF*, 2024, URL <https://rdflib.dev/>. Online; (Accessed 26 June 2024), Version 6.2.1.
- [86] A. Dimou, M. Sande, P. Colpaert, R. Verborgh, E. Mannens, R. Walle, *RML: A generic language for integrated RDF mappings of heterogeneous data*, 2014, URL <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f0b98c4fc3a542a83349666f4073359ed56d1a17>.
- [87] P. Aivaliotis, K. Georgoulis, Z. Arkouli, S. Makris, Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance, *Procedia CIRP* 81 (2019) 417–422, <http://dx.doi.org/10.1016/j.procir.2019.03.072>, URL <https://www.sciencedirect.com/science/article/pii/S2212827119303774>. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12–14, 2019.
- [88] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. O'Riain, *Linking building data in the cloud: Integrating cross-domain building data using linked data*, *Adv. Eng. Inform.* 27 (2) (2013) 206–219, <http://dx.doi.org/10.1016/j.aei.2012.10.003>, URL <https://www.sciencedirect.com/science/article/pii/S1474034612000961>.