Towards Automation in Steel Construction: Development of an OWL Extension for the DSTV-NC Standard

V. Jung[a], **L. Kirner**[a] **and S. Brell-Cokcan**[a]

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

E-mail: jung@ip.rwth-aachen.de, kirner@ip.rwth-aachen.de, brell-cokcan@ip.rwth-aachen.de

Abstract –

The automation of steel construction requires new possibilities to link process data, measured deviations and tolerances. Our current research in robotic steel fabrication aims to address this challenge by developing an adaptive information model interface that can seamlessly integrate the cross-process considerations necessary for precise and efficient fabrication. The objective is to further develop existing information interfaces and to increase the use of flexible and partially automated robot concepts in steel construction. Our approach builds on existing standards and product interfaces e.g., DSTV-NC in steel construction and focuses on converting and extending them with help of an ontology including tolerances and process parameters. The results provide a contribution to the development of automated systems in construction and promote small and medium-sized enterprises in steel construction to deal with current challenges of skill shortages, productivity, and occupational safety.

Keywords –

Domain Ontology, Robotic Steel Construction, Linked Data, Semantic Web

1 Introduction

In today's construction industry, (partially) automated production systems are used almost exclusively for prefabrication processes. However, in steel construction, several characteristics make the widespread integration of automated production systems difficult [1]. The main challenges include geometric tolerances, material variations, component size and weight, as well as small batch sizes of complex assemblies. In comparison to other industries, steel construction has significantly higher component and manufacturing tolerances [2–5].This hampers the integration of robots because path planning is usually based on ideal component dimensions. For the widespread use of robots, there is a lack of a transparent and continuous information process as well as a standard for the storage of the deviation and process information of the real component deviations in an adaptive information model. Currently, some machines are already capable of measuring the real dimensions of single parts but there is no defined place to feedback the information to the virtual information model. In addition, a connection between planning and fabrication data is missing, which would also increase the possibility to optimize and analyze the processes based on real data.

Full robotic processes require access to machine readable data, resulting in a demand to digitalize steel tolerance norms. An efficient interface would enable machines to exchange tolerance and process information and could therefore help to implement concepts according to Industry 4.0 [6].

Therefore, the motivation of this research is the development of an ontology (see chapter 2.2) which enables the use of semantic web technologies for the description of steel construction information. This includes process information needed for robotic processes, manufacturing process metadata such as tolerances and data feedback including quality measurements. Identifying and capturing critical deviations as well as resource-bound process parameters could raise efficiency and enable robotic workflows. In addition, further opportunities for process optimization can be developed.

The approach of using semantic web technologies to describe and link plan data with real fabrication and process information in steel construction was chosen because the technology promotes a continuous flow of information and allows all stakeholders to be connected, regardless of their software or machines. The concept will be verified in a real demonstrator to show the feasibility.

2 Related Work

The manufacturing process of steel construction is predominantly carried out in individual and small series production [7]. It is divided into planning, production and assembly. In practice, there is a spatial separation between the three process steps mentioned. Due to the lack of back documentation, there is no central information model that combines the data of all involved actors. In principle, graphic or product interfaces can be used to transmit information about components. Graphic interfaces describe the products in geometric form; however, a description of how these can be produced is not included. The product interface DSTV-NC format was developed by “bauforumstahl” and is currently the most widely used data format [8, 9].

2.1 DSTV-NC standard as a base for robotic manufacturing

The DSTV-NC standard in steel construction represents a fundamental interface between design, production planning and machine control. It functions as an interface between CAD/CAM applications and NC production in steel construction. The DSTV-NC standard allows the control of various NC machines via the respective postprocessing. In particular, subtractive manufacturing processes using sage drilling machines, flame-cutting and punching machines as well as 5-axis CNC machining are used. Based on that format a prototypical interface for the control of an industrial robot was implemented by developing an independent plugin for Grasshopper3d. Programming itself represents a great barrier in using robots for individual tasks and small batches as knowledge and special skills are required to adapt the path planning to the current task. To overcome this problem, an efficient task-oriented approach was chosen for the implementation of the interface, in which a task corresponds to a machining step according to DSTV-NC. For these tasks, a strategy was stored in the interface, with which robot code is automatically generated. For collision-free machining of several sides of a component, a rule-based global path planning was developed and implemented [10].

During the validation it was highlighted that the DSTV-NC format does not contain real part information, as the information refer to the planned geometry. Due to the specific characteristics of steel and the large part sizes there are deviations from the planned geometry to the real dimensions of the steel part. This information is missing so far in the DSTV-NC format, consequently the path plan must be manually adapted to avoid any collision of the robot with the part. The modular structure of this format could allow the extension with additional information. So far, however, it is only used as a manufacturing instruction with ideal geometries, without the possibility of feedback of real values or updating values [11].

Currently, another group is focusing on transferring the information from the DSTV-NC standard into the Industry Foundation Classes (IFC), which is the common exchange format for the Building Information Modelling (BIM) data. The objective is to be able to operate and generate machine code based on an IFC description of the desired work piece. IFC could be an alternative for storing additional tolerance and process information to enable a robotic process for steel construction. IFC forms the basis for an open, neutral and standardized data format and, in addition to the 3D data model, contains extensive data structures for describing objects from almost all sectors [11, 12].

2.2 Ontologies as a base for robotic manufacturing

Ontologies seem to be a promising solution for semantic interoperability problems [13–16]. In Computer Science an ontology is defined as "an explicit specification of conceptualization for a domain of interest" [17]. Typically, a logical theory written in a particular language formalizes the conceptualization within an ontology. Ontologies also provide reasoning capabilities that can be used to infer new knowledge. Ontologies contain explicit and unambiguous data semantics, and thus enable semantic interoperability, because they contain a formal representation of concepts, individuals and the relationships between these concepts, data and entities [18].

In recent years, many ontologies have already been developed for the domain of construction. Most approaches are based on the translation of already existing models [19, 20] or the development of new mapping techniques [21]. The previous achievements in this area mostly do not describe steel construction processes but approaches for the mapping of general geometric and topological information or the description of product and component information [21, 22].

In various research projects such as Digitizing Construction Workflows (DiCtion) or the Internet of Construction (IoC) ontologies for describing general construction processes have been developed [23, 24].

The concept of the IoC ontology is built around the core concept of ioc:process. This top level ontology is meant to create an interconnection to different sub-domains of construction, such as steel construction [24]. Previous works that focused to form domain ontologies for the steel construction are scarce. To enable a global optimization of the overall steel production value chain, Zillner et al. aimed to address the challenges of the information exchange between boarders, locations and companies [25]. There are also specific ontologies for robotics, such as OCRA [26] but they do not focus on the domain of steel construction and are therefore not in scope of this paper. In the future, it may function as a top-level ontology for approaches such as the *DSTV* ontology.

However, the advantages of using ontologies for process data have not yet been addressed nor developed. Established concepts like DSTV-NC format can be easily integrated into the Semantic Web stack. This enables to link heterogenous and unstructured data and includes various sources of information, such as BIM or scheduling data.

3 Methodology

This section describes the workflow which was executed to transfer the existing information from the DSTV-NC standard to an ontology created in Web Ontology Language (OWL), including the extension of specific information which are not yet defined in the DSTV format but required for a robotic manufacturing process. The ontology is referenced in the following as "DSTV ontology" (*dstv*) with the Base URI set to http://ip.rwth-aachen.de/dstv#.

3.1 Ontology Design

Unlike the conventional methods of ontology creation [27], the DSTV ontology was initially developed by aligning the structure of the XML-based standard with the concepts and properties in OWL. Table 1 shows an excerpt of the DSTV Classes (BFS-RL 03-105) [9].

Table 1. Extract tolerances for position of screw holes

DSTV Class	Description, "german description"
hljob	hole job, "Durchgangsloch"
stjob	screw thread, "Gewinde"
bhjob	blind hole, "Sackloch"
shjob	sink hole, "Senkung"
ohjob	oblong hole,"Langloch"
pmjob	punchmark,"Einzelkörnung"

The information contained in the DSTV-NC was restricted to workpiece information and details related to drilling processes, specifically those defined as *hl*, *hljob*, etc. As a result, the corresponding OWL ontology comprised 258 classes and 1583.

Based on the analysis which information, processes and resources are required for the robotic manufacturing process, new concepts, and classes for the extension of the current DSTV-NC model are defined. These concepts focus on the tolerance information defined in DIN-norms, measurement concepts for different steel profiles as well as models for storing process and machine data. Alignments to the IoC ontology as well as the ifcOWL and Linked Building Data (LBD) ontologies are added. As seen in fig 1, this enables to link design and fabrication data-sources as well as dynamic process information.

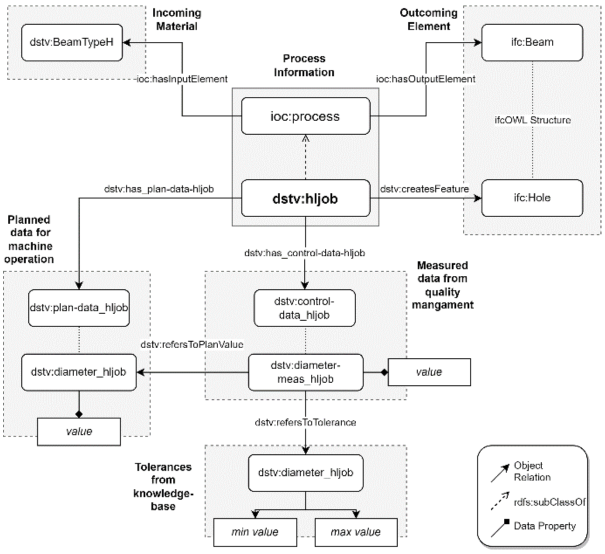

Figure 1: Abstract overview over the DSTV ontology with aligning concepts of ifcOWL and ioc ontology

By facilitating the alignment of planning data from DSTV-NC to IFC, the suggested ontology can ensure that both formats are consistent with each other. This alignment is particularly important for data such as diameter, as illustrated in Figure 1, which must match in both formats.

The proposed ontology can be utilized even in the absence of an IFC file, which is especially crucial when domain-specific software generates DSTV-NC but cannot be integrated with IFC. As the depicted structures are overly intricate to be represented coherently in the graphic, the object properties within the boxes have been simplified.

3.2 Extension of tolerance information

To display and collect all the relevant information for the steel construction process an extensive research and personal discussion with steel fabrication experts have been undertaken. Besides the status quo of the current information flow processes and the used data formats it was identified that some machines are already capable of measuring the part before and/or after it was processed. It is lacking a place to store the measured data and integrate defined tolerances. Tolerances are essential in defining the allowable variations in dimensional and geometric parameters of steel components such as beams, columns and connections. They are used to ensure that the different parts of a structure fit together correctly during assembly to provide the desired function, stability and performance under load conditions.

Manufacturing tolerances of steel sections describe the accepted deviations of the steel element itself, such as profile height, flange width, web thickness or flange thickness. Depending on the steel section, the standardized values for these tolerances can be found in relevant building codes, standards and specifications. In particular, DIN EN 10034 (I and H beams), DIN EN 10279 (U beams), DIN EN 10219-2 (cold-rolled hollow sections) and DIN EN 10210-2 (hot-formed hollow sections).

At present, the tolerance specifications for the various processes in steel construction are only human readable.

The process of integrating these tolerances into an ontology started with an Excel spreadsheet to collect the information from the various standards. This included all relevant tolerance types, the standard to which they belong, the values and the conditions under which they can be applied. For transfer into the ontology, manufacturing tolerances are first distinguished by steel sections, which then list all relevant tolerances for a specific steel section. Process tolerances are similarly subdivided with the drilling process as the central class for all tolerances associated with that process. Table 2 displays an extract of the defined tolerances for a bore process, here the position of a screw hole according to DIN EN 1090-2.

Table 2. Extract tolerances for position of screw holes

Parameter	Basic tolerance	Extending tolerance class 1	Extending tolerance class 2
Deviation of the center line	+2,0/-2,0 [mm]	+2,0/-2,0 [mm]	+1,0/-1,0 [mm]
Deviation of the distance "a" between a single hole with diameter "d" and the edge of a sheet metal			
if a < 3d	+0,0/-0,0 [mm]	+3,0/-0,0 [mm]	+2,0/-0,0 [mm]
if a ≥ 3d	+3,0/-3,0 [mm]	+3,0/-3,0 [mm]	+2,0/-2,0 [mm]

In the case of determining the tolerance for the position of a screw hole, the *dstv:hljob* is linked to a measured value for the *vertex-x* and *vertex-y* on the relevant surface through *dstv:has_control-data-hljob* and its subcategories. Subsequently, these values are matched against the applicable tolerance value, which varies depending on the relevant tolerance classes defined in specific DIN standards. The tolerance node is connected to the *control-data-hljob* by the object property *dstv:refersToTolerance*. Once these values are aligned, the modeled tolerances can be easily connected to every process that belongs to the dstv: classes such as *dstv:hljob*.

3.3 Extension of process information

In addition to tolerance information, process data must also be stored. This is mainly done via *ioc:process* and its aligned concepts. Certain concepts in the proposed ontology have areas of overlap with the general process concepts outlined in DSTV-NC. This is the case for attributes such as the machine name, manufacturer, process speed, and the tool used, along with its respective properties. However, by utilizing the relevant ontologies, the proposed framework also enables the management of dynamic information that emerges during the robotic process execution. Consequently, feedback data pertaining to the starting and ending times of various manufacturing steps, in the form of timestamps, can be incorporated into future datasets.

4 Use-Case

This section provides an illustration of how the created capabilities can be leveraged to map manufacturing data in steel construction, using the DSTV ontology as a guide. (see fig. 2).

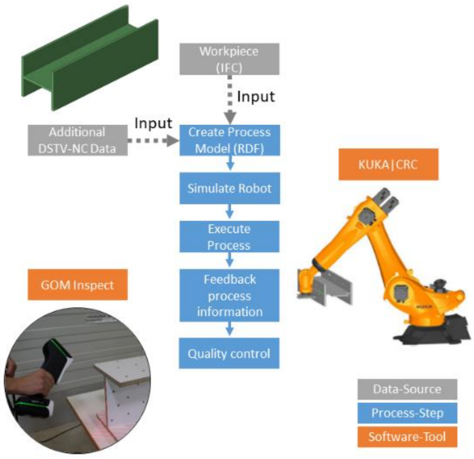

Figure 2: Coarse depiction of the use case including data sources, process steps and software tools.

We propose an approach using an IFC file as the source material and existing approaches for Linked Building Data [28]. For the robotic process, tools and interfaces are used that are developed by robots in architecture [29] and are available to us as WIP - in development versions.

4.1 Model conversion

The example manufacturing process is based on an

IPE 300 profile that is to be drilled with two holes. The model of the profile has been created and provided by an ad-hoc workgroup for ongoing research into the implementation of the DSTV-NC logic in the IFC data model.

The supplied file is converted to Resource Description Framework (RDF) using the IFC2LBD [30] converter which also includes conversion via the ifcOWL ontology (IFC4_ADD2_TC1) [31]. For storing the created triples, an instance of the Stardog triple store is installed on the university server. The triple store provides a SPARQL endpoint that can be used to add, modify and query the linked data in the graph. The original IFC4 file, which is 48KB in size, corresponds to approximately 6100 triples.

4.2 Process data modeling

In a next step, the processes for the robotic fabrication are instantiated using the ioc process ontology (http://w3id.org/ioc#). The ontology is subject to unpublished research as part of the Internet of Construction project [24]. The processes consist of a parent process that link the input element (IPE300) with the information of the output element (the specific profile including two drilled bores). The detailed child processes correspond to the intended robot movements. As example five movements were created, that are initialized with the following statements:

```
inst:safeto_hljob_01 a ioc:Process.                (1)
inst:hljob_01 a ioc:Process, dstv:hljob.           (2)
inst:traveltohljob_02 a ioc:Process.               (3)
inst:hljob_02 a ioc:Process, dstv:hljob.           (4)
inst:safetoEndpos a ioc:Process.                   (5)
```

The processes are then put in a sequence via *ioc:hasSucessor* and *ioc:hasPredecessor* statements. The drilling processes in (2) and (4) are belonging to both the *ioc:Process* and *dstv:hljob* class. For the cause of better readability, the prefixes are not stated here. The example is further shown via a detailed description of the first drilling process (2).

```
inst:hljob_01 a ioc:Process, dstv:hljob;           (6)
ioc:hasPredecessor inst:safeto_hljob_01;           (7)
ioc:hasSuccessor inst:traveltohljob_02;            (8)
ioc:hasMethod inst:iocMethod_02;                   (9)
dstv:has_control-data-hljob                        (10)
inst:hljob_01_control-data;
dstv:has_fab-data-hljob inst:hljob_01_fab-data;    (11)
dstv:has_plan-data-hljob inst:hljob_01_plan-data;  (12)
dstv:has_drilling_tolerances                       (13)
inst:drilling_tolerances_diameter_01;
dstv:refersToIfcElement                            (14)
inst:IfcVoidingFeature_195.
```

The developed processes serve as a means of integration, enabling the incorporation of information extracted from the DSTV-NC standard, as well as supplementary metadata, including tolerances and deviations. The drilling processes are classified as *rdf:type dstv:hljob* (6), a classification that stems from the original process descriptions outlined in the DSTV-NC standard. Using the object property *dstv:refersToIfcElement* (14), the processes are connected to the IfcVoidingFeature that they generate. Planning, fabrication, and quality control data related to DSTV-NC are linked with object properties *dstv:has_plan-data-hljob*, *dstv:has_fab-data-hljob*, and *dstv:has_control-data-hljob*, respectively (10-13). Generating data for some of these nodes requires manual input since the IFC File does not include all of the necessary parameters, although data such as hole diameter and position can be sourced from the IFC file. For specific control parameters, such as diameter tolerances, the corresponding nodes are linked to the modeled values via object properties *dstv:refersToTolerance* and *dstv:refersToPlanValue*.

In this stage, tolerances are simply modeled by inserting limit sizes and value ranges. For the diameter tolerance, the values were chosen from DIN standards such as DIN EN 10034 and described as min and max value via data properties.

4.3 Path planning

For the shown exemplary implementation, the automated path planning for the robot was done in Rhino / Grasshopper with the KUKA | crc plugin. To retrieve the data, a WebSocket was used to query the database. For this, three SPARQL queries retrieve planes, the geometric information of the IPE profile and the geometric information of the holes. Besides advanced logic to prevent collisions between the robot and the workpiece, only the query for the planes is strictly needed to generate the path. In (15-17) excerpts from the SPARQL query for retrieving planes are shown. The full query consists of 54 lines including prefixes. The shown part of the script solely queries the cartesian plane-basepoint.

```
?void ifc:predefinedType_IfcVoidingFeature           (15)
ifc:HOLE.
?void ifc:objectPlacement_IfcProduct ?place.         (16)
?place ifc:coordinates_IfcCartesianPoint ?carts.     (17)
```

Querying the geometry was added as it is a convenient way to supervise the algorithms in the Rhino Environment as 3D geometry can be rebuild from it. To do this simple python scripts were implemented in ghpython. These recreate the geometry from the query response which is given in the SPARQL 1.1 Query Results JSON Format recommended by the W3C.

Semantic description of the required planes has a benefit that enables the robot's movements to correspond to specific commands based on the process class. With IoT connectivity, tools can automatically trigger specific actions (in this case, drilling) when a particular plane is connected to a *dstv:hljob*. Related research describes workflows to accomplish this task. [32]. The use of this logic enables the generation of a path planning that includes tool control. KUKA|crc stores this in a JSON object (18), which can be added to the database via SPARQL UPDATE.

```
{
    "id": "a3c478c8",
    // Unique Command Identifier for Information
Flow Tracing
    "cmd": "MovePtp",
    // Basic Command Typ or Skill Class Name
    "dev": "robotid",
    // Executing Device the Command is Fowarded
to
    "sync": false,
    // Synchronization Flag for Inter-device
Sychonization
    "tcpPose": {
        "X": 500.0,
        "Y": 500.0,
        "Z": 420.0,
        "A": -180.0,
        "B": 0.0,
        "C": 180.0
    },
    /* Command Class Specific Parameters such as
     * TCP Target Position, Axis Target Position
etc.
     */
    "meta": {
        "status": "010",
        "redAxis": 45.0,
        "speed": 80.0
    }
```

(18)

The objects that comprise all the necessary information for the robotic process are linked to the modeled processes in the graph through a *ioc:hasMethod* and a *schema:value* property.

4.4 Robotic manufacturing

When production is due, a simple python script running on a WebServer, automatically queries the database to check if all necessary information are available, as well as checking the status of robot and tool. If ready, the JSON object, which includes the robot control information, gets send to a KUKA|crc instance connected to the robot. This is done via the MQTT protocol. The robot then interprets the commands and starts the process sequence. The test setup is shown in fig. 3.

After every process (-step), the robot sends a command to the WebServer to adjust the process status of the given process. The timestamps generated can later be used to optimize the movement and thus time consumption of the process.

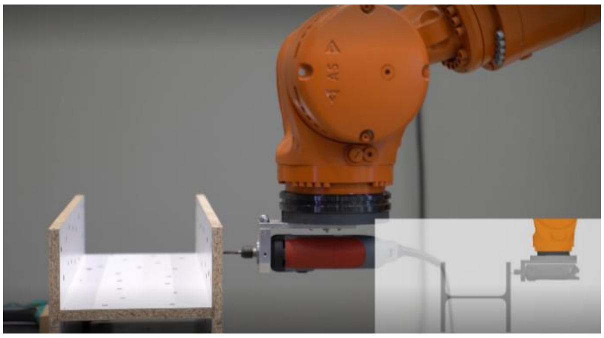

Figure 3: Robotic fabrication test with a wooden dummy IPE 300 profile. Process parameters are described by the DSTV ontology and accessed via MQTT.

4.5 Quality control and feedback of information

Figure 4: Hand held 3D scanner which was used to capture deviations

After the bore holes were robotically drilled, the part was scanned with a 3D hand-held laser scanner (see fig. 4). The scanned point cloud was analyzed with the GOM Inspect software.

The measured data can be added to the overall semantic web description via the python interface of GOM and a SPARQL UPDATE Query. In the modeled graph the data is connected via the *dstv:has_control-data-hljob* object property. As the model has also access to the defined tolerances of the drilled hole and the planned geometry data a comparison between these values can be executed. The result of the used query returns a JSON object. It shows that the control value for the x-vertex is 15.24mm (19) while the plan value is 15.0mm (20)

```
                "controlvertex_x": {
                    "datatype":
"http://www.w3.org/2001/XMLSchema#float",
                    "type": "literal",
                    "value": "15.24"
```

(19)

```
            }
            "planvertex_x": {                         (20)
                "datatype":
  "http://www.w3.org/2001/XMLSchema#float",
                "type": "literal",
                "value": "15.0"
            }
```

The resulting graph of the whole use-case consists of 6400 Triples. Figure 5 shows the resulting interconnected data model, based on the *ioc:process* in the center, which links the IFC based data (incoming material and created features – here *ifcHOLE*) and the specific DSTV extension data. It is visible that the DSTV extension consist of three branches which are the planned and measured data as well as the pre-defined tolerances.

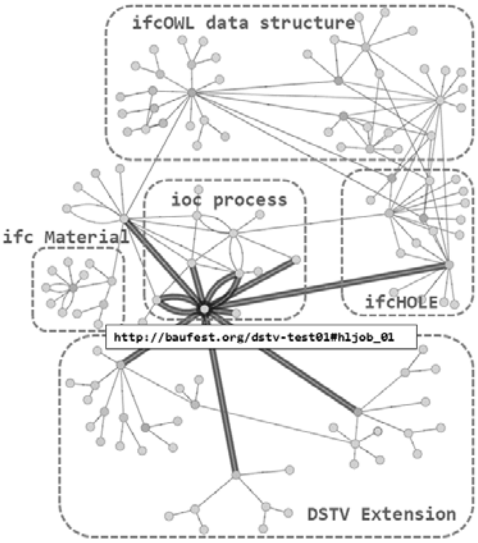

Figure 5: The resulting graph of the use-case

5 Outlook

The DSTV ontology allows the use of semantic web technologies for the description of steel construction information. It enables the description of robotic process and overall steel construction information and connects all the relevant information of different fabrication processes, including manufacturing process metadata such as tolerances and data feedback. The additional information can be used to automate the evaluation of the measured data, as well as to analyze which tools cause which deviations from the planned geometry. Optimization of processes based on real data is feasible and promotes more efficient steel construction processes.

The first draft of the DSTV ontology was able to demonstrate that the concept is feasible and easy to implement, as it follows the structure of the DSTV-NC from which it originates, extending it only where necessary. However, future research should evaluate the ontology further to optimize and simplify it. It should also address the limitations and prerequisites of this work such as the required digital data and modeling expertise or solutions to current disadvantages of using the IFC data model such as data loss or version conflicts.

6 Acknowledgements

The project (BauFeSt 4.0) (21690 N/FE1) is being carried out in cooperation with the industrial partners and financially supported by the Federal Ministry for Economic Affairs and Energy (BMWI) via the German Federation of Industrial Research Associations (AiF). This work is also part of the research project Internet of Construction (funding number 02P17D081).

7 References

[1] *Kuntze, P.; Mai, C.-M.:* Arbeitsproduktivität - nachlassende Dynamik in Deutschland und Europa. *In:* WISTA - Wirtschaft und Statistik 72 (2020), Heft 2, S. 11-24.

[2] *Sikstrom, F.; Ericsson, M.; Nylen, P. et al.:* 3D-Scanning for Weld Distortion Measuring. *In: :* Proceedings of the 23rd IEEE Instrumentation and Measurement Technology Conference – IMTC/06 ; Sorrento, Italy - 24-27 April 2006. IEEE Instrumentation and Measurement Society. Inst. of Electrical and Electronics Engineers, Sorrento, Italy, 2006, S. 2132-2137.

[3] *Wilmsmeyer, H.; Flasskamp, M.; Schierenberg, M.-O.:* Intelligente Korrektur eines Schweißroboters. *In: Jasperneite, J.; Lohweg, V.* (Hrsg.): Kommunikation und Bildverarbeitung in der Automation, Technologien für die intelligente Automation. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018, S. 283-294.

[4] DIN EN 1090-2: Ausführung von Stahltragwerken und Aluminiumtragwerken - Teil 2: Technische Regeln für die Ausführung von Stahltragwerken. Norm, Ausgabe 2018.

[5] *H. Wilmsmeyer; M. Flasskamp; T. Jungeblut et al.:* Intelligenter Schweißroboter: Selbständige Reaktion eines Schweißroboters auf große Toleranzen. *In:* wt Werkstattstechnik online (2016), Heft 106, S. 595-599.

[6] *Plattform Industrie 4.0:* Digitale Ökosysteme global gestalten – Leitbild 2030 für Industrie 4.0 (2019).

[7] *R. Fischer:* Prozess- und Modellorientierung des Stahlbaus: Ein Beitrag aus der Praxis. *In: Motzko, C.* (Hrsg.): Praxis des Bauprozessmanagements – Termine, Kosten un Qualität zuverlässig steuern. Ernst & Sohn, Berlin, 2013, S. 193-212.

[8] *bauforumstahl e.V.:* Standardbeschreibung von Stahlbauteilen für die NC-Steuerung (ASCII), 2003.

[9] *bauforumstahl e.V.:* Standardbeschreibung von Stahlbauteilen für die NC-Steuerung (XNC), 2006.

[10] *Jonas Hippe:* Entwicklung einer prototypischen Schnittstelle zur Ansteuerung von Industrierobotern über den DSTV-NC. Aachen, RWTH, Masterarbeit, 2021.

[11] bauforumstahl news. *In:* Stahlbau 87 (2018), Heft 2, S. 173-190. https://doi.org/10.1002/stab.201870214.

[12] DIN EN ISO 16739-1 (2021-11-00): Industry Foundation Classes (IFC) für den Datenaustausch in der Bauwirtschaft und im Anlagenmanagement - Teil 1: Datenschema. Norm, Ausgabe 2021.

[13] *Sabou, M.; Biffl, S.; Einfalt, A. et al.:* Semantics for Cyber-Physical Systems: A cross-domain perspective. *In:* Semantic Web 11 (2020), S. 115-124. https://doi.org/10.3233/SW-190381.

[14] *Wong, A.; Ray, P.; Parameswaran, N. et al.:* Ontology mapping for the interoperability problem in network management. *In:* IEEE Journal on Selected Areas in Communications 23 (2005), Heft 10, S. 2058-2068. https://doi.org/10.1109/JSAC.2005.854130.

[15] *Gyrard, A.; Zimmermann, A.; Sheth, A.:* Building IoT based applications for Smart Cities: How can ontology catalogs help? *In:* IEEE internet of things journal, Vol. 5 (2018), Iss. 5, pp. 3978-3990. https://doi.org/10.1109/jiot.2018.2854278.

[16] *Cao, Q.; Giustozzi, F.; Zanni-Merk, C. et al.:* Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach. *In:* Cybernetics and Systems 50 (2019), Heft 2, S. 82-96. https://doi.org/10.1080/01969722.2019.1565118.

[17] *Gruber, T.R.:* A translation approach to portable ontology specifications. *In:* Knowledge Acquisition 5 (1993), Heft 2, S. 199-220. https://doi.org/10.1006/knac.1993.1008.

[18] *Patil, L.; Dutta, D.; Sriram, R.:* Ontology-Based Exchange of Product Data Semantics. *In:* IEEE Transactions on Automation Science and Engineering 2 (2005), Heft 3, S. 213-225. https://doi.org/10.1109/TASE.2005.849087.

[19] *buildingSMART Technical:* ifcOWL - buildingSMART Technical, 2023, https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/ [Zugriff am: 05.01.2023].

[20] *Mathias Bonduel; Jyrki Oraskari; Pieter Pauwels et al.:* The IFC to Linked Building Data Converter-Current Status. *In:* , 2018.

[21] *BOT: the Building Topology Ontology of the W3C Linked Building Data Group:* swj2224.

[22] *Wagner, A.; Sprenger, W.; Maurer, C. et al.:* Building product ontology: Core ontology for Linked Building Product Data. *In:* Automation in Construction 133 (2022), S. 103927. https://doi.org/10.1016/j.autcon.2021.103927.

[23] *Zheng, Y.; Törmä, S.; Seppänen, O.:* A shared ontology suite for digital construction workflow. *In:* Automation in Construction 132 (2021), S. 103930. https://doi.org/10.1016/j.autcon.2021.103930.

[24] *Kirner, L.; Lublasser, E.; Brell-Cokcan, S.:* Internet of Construction: Research Methods for Practical Relevance in Construction. *In:* Technology|Architecture + Design 5 (2021), Heft 2, S. 146-152. https://doi.org/10.1080/24751448.2021.1967053.

[25] *Zillner, S.; Ebel, A.; Schneider, M.:* Towards intelligent manufacturing, semantic modelling for the steel industry. *In:* IFAC-PapersOnLine 49 (2016), Heft 20, S. 220-225. https://doi.org/10.1016/j.ifacol.2016.10.124.

[26] *Olivares-Alarcos, A.; Foix, S.; Borgo, S. et al.:* OCRA – An ontology for collaborative robotics and adaptation. *In:* Computers in Industry 138 (2022), S. 103627. https://doi.org/10.1016/j.compind.2022.103627.

[27] *Natasha Noy:* Ontology Development 101: A Guide to Creating Your First Ontology (2001).

[28] The Linked Building Data Community Group, 2022, https://w3c-lbd-cg.github.io/lbd/ [Zugriff am: 31.01.2023].

[29] Association for Robots in Architecture – making robots accessible to the creative industry, 2023, https://robotsinarchitecture.org/ [Zugriff am: 31.01.2023].

[30] *M. Bonduel; J. Oraskari; P. Pauwels et al.:* The IFC to linked building data converter - current status (2018).

[31] *Pieter Pauwels; walter.terkaj@itia.cnr.it.* OWL ontology for the IFC conceptual data schema and exchange file format for Building Information Model (BIM) data [online], 2019 [Zugriff am: 31.01.2023], https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL/index.html.

[32] *Dai, R.; Kerber, E.; Reuter, F. et al.:* The digitization of the automated steel construction through the application of microcontrollers and MQTT. *In:* Construction Robotics 4 (2020), 3-4, S. 251-259. https://doi.org/10.1007/s41693-020-00042-9.