dstv: An ontology-based extension of the DSTV-NC standard for the use of linked data in the automation of steel construction

Lukas Kirner, Jyrki Oraskari, Victoria Jung, Sigrid Brell-Cokcan

LDAC 2023 - 15.06.23

individualized production | RWTH AACHEN UNIVERSITY

Motivation

Video 01: Example for state of the art automated steel assembly: ZEMAN's "SBA2 Conti+„ [1]

Oberservation 1:

Automated assembly requires very tight tolerances both in material and prefabricated components.

- economic viability suffers

Oberservation 2:

Many production machines have inbuilt metrology but their information system is closed / encapsuled.

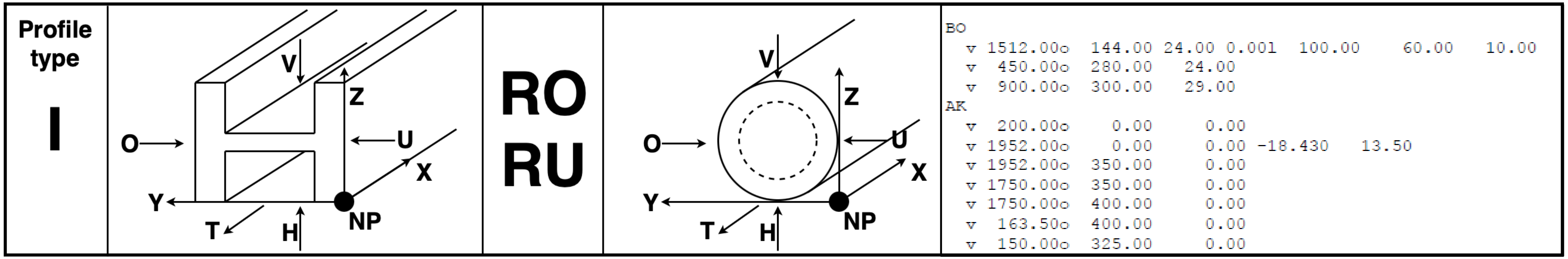
- Metrology data can not be used outside of the machine

Approach:

Extension of an already used format / standard by metrology data to allow machines to adapt their processes accordingly.

DSTV-NC

DSTV-NC: A standard interface of steel construction part geometries for NC manufacturing operations



Advantages:

- ✓ Commonly used in steel industry
- ✓ Widely used version (ASCII) easy to implement
- ✓ Process driven design
- ✓ „Well-defined“ / „clear“

Shortcomings:

– Cannot store tolerances
– Cannot store measurements

– Not linked to resulting building product
– Not linked to machine information

– No dynamic information
– Standard itself not machine-readable

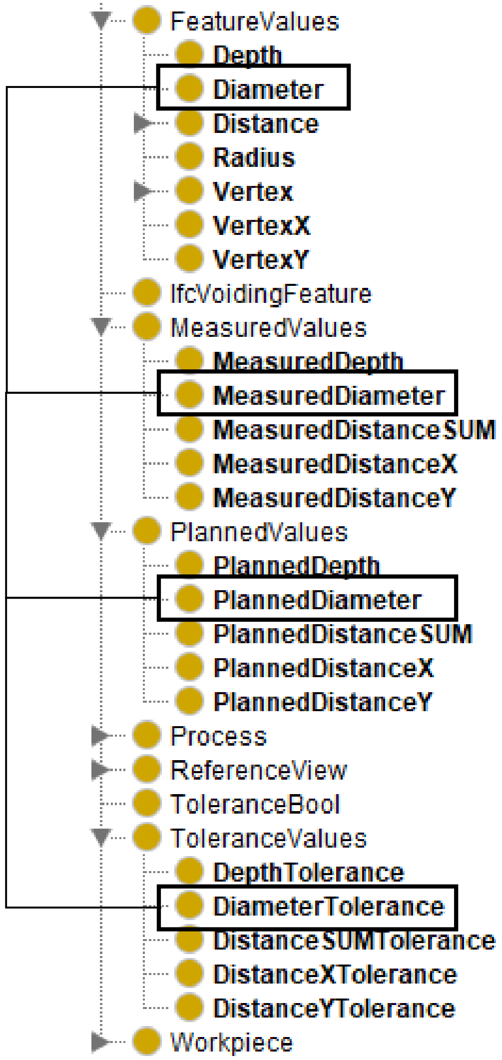
Ontology Design

Methodology according to „Ontology Development 101“ by Noy and McGuinness [2]

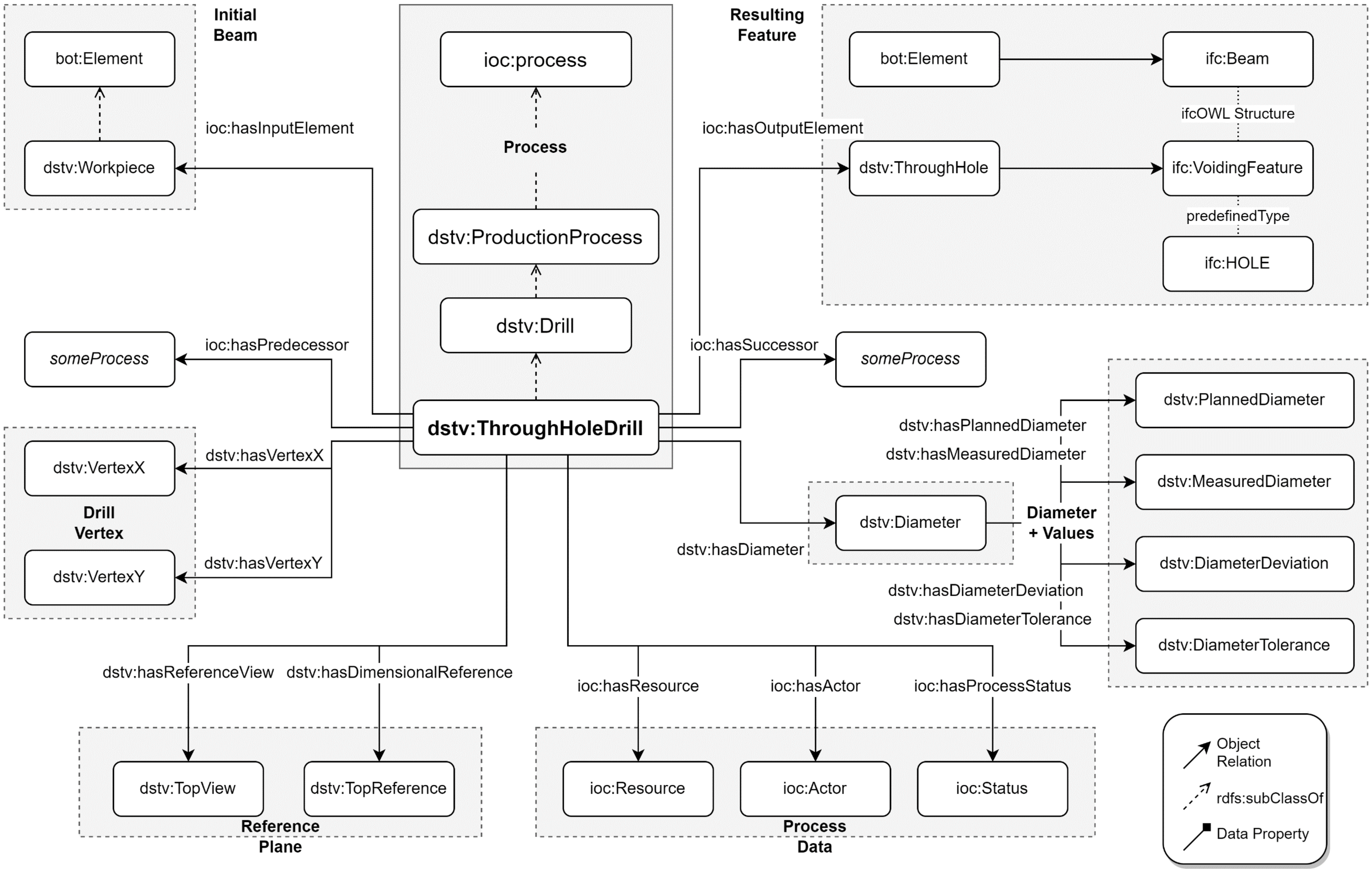
- Formulation of 10 „simple“ competency questions (CQ)
- No specific Ontology for the domain of „**automated steel manufacturing**“ available
- Creating concepts directly aligned to DSTV-NC standard

Specific Features

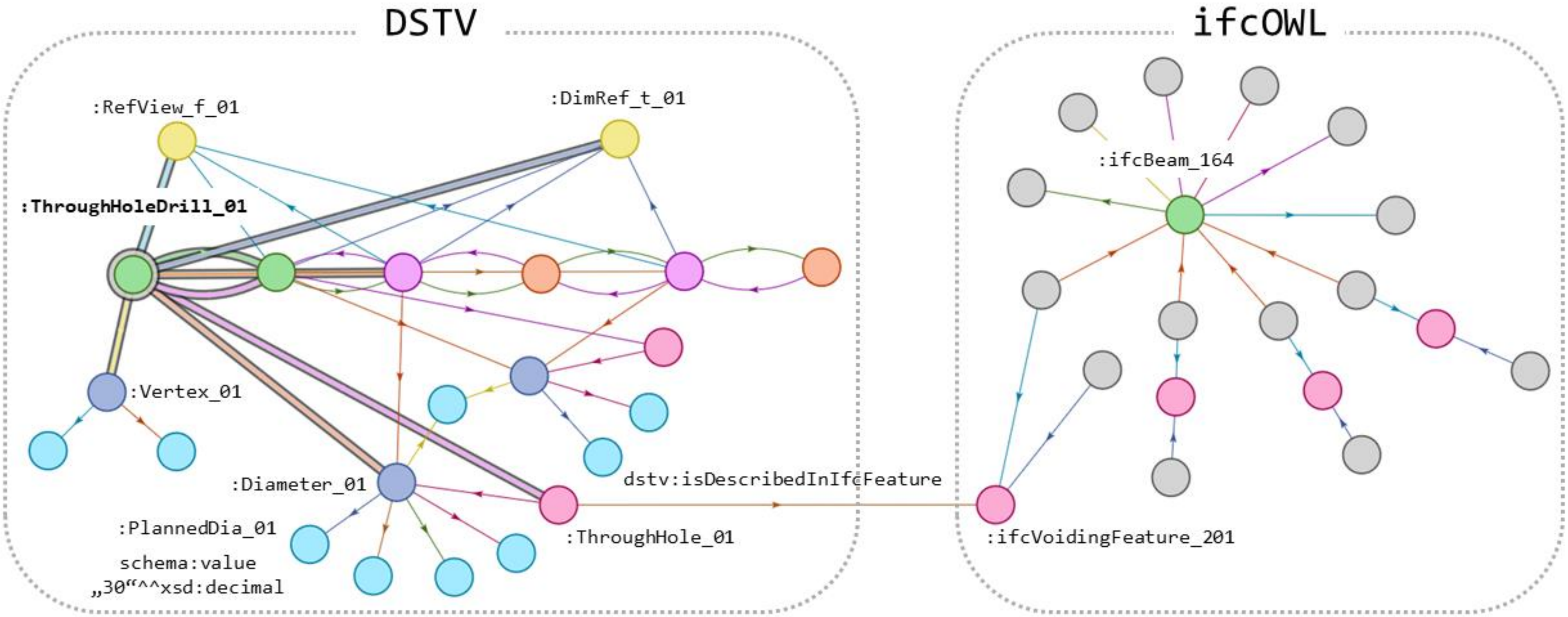
- Process-driven design (feature values linked to processes)
- Works with and without *LBD/ifcOWL* concepts
- Structure for „planned“, „measured“, „deviation“, … duplicated
- Versioning / dynamic features according/inspired by *OPM* ontology [3]



Example: Drill



Example: Drill

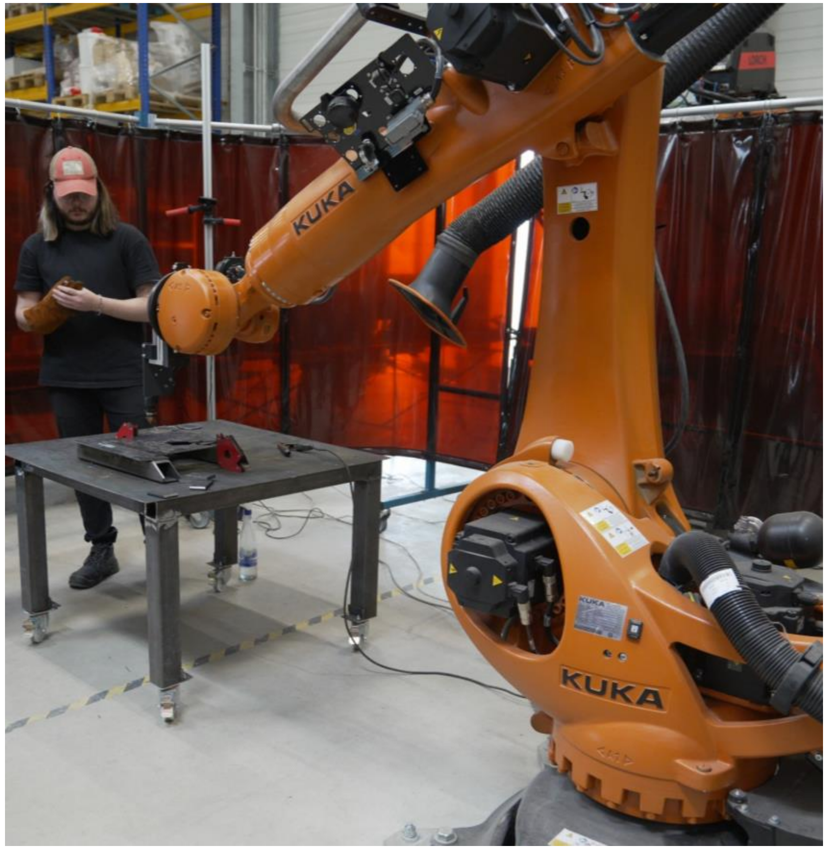


Process-Scenario: Plasma-Cutting an IPE 220

Process: **Plasma Arc Cutting**
Workpiece: IPE 220
Machine: KUKA KR240 R2700 „optimus“
Tool: Hypertherm Powermax

Constraints / Parameters:

Torch	status	on/off
Torch	distance	(mm)
Robot	speed	(mm/s)



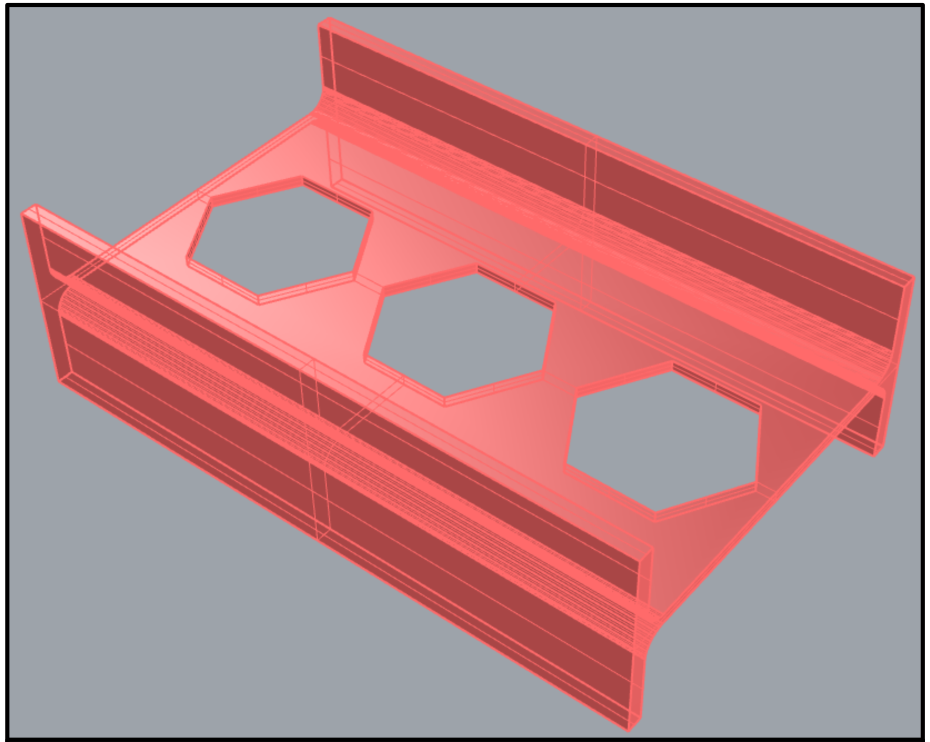
Process Sequence:

Travel from save to first cut → **First cut** → **Travel** from first to second cut → **Second cut** → **Travel** from second to third cut → **Third cut** → **Travel** to EndPOS

Sequence – Step 1: Workpiece Modeling

- **Create workpiece**
  - Model as IFC4 (SPF)
  - IFCtoLBD including ifcOWL
  - SPARQL UPDATE

- **Generate *DSTV* planning data**
  - Query Geometry: SPARQL SELECT
  - Define Reference Plane
  - Generating RDF-Lists of Vertices
  - SPARQL UPDATE

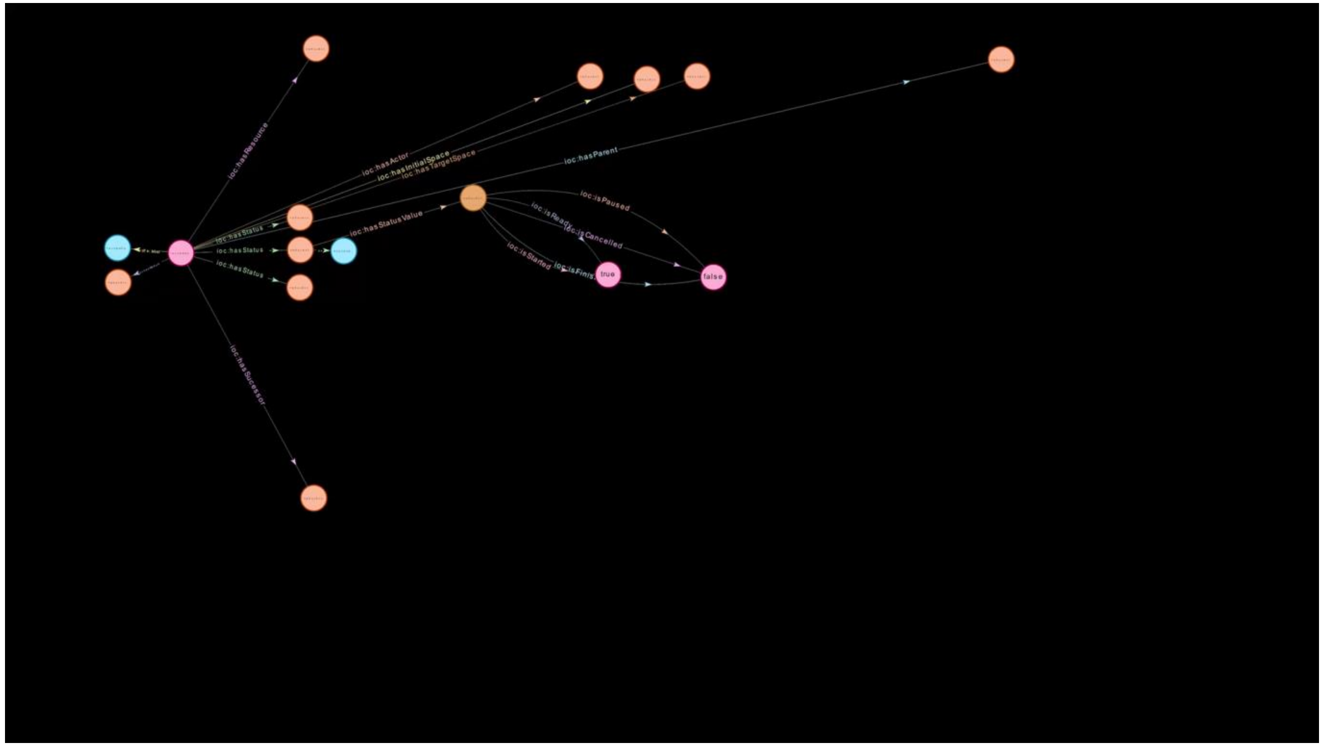


```
#PlannedVertexX_01_1> schema:value "110.0"^^xsd:double.
#PlannedVertexY_01_1> schema:value "-10.5"^^xsd:double.
#PlannedVertexZ_01_1> schema:value "0.0"^^xsd:double.
```

Sequence – Step 2: Process Modeling

- **Creating a process-sequence using the *ioc: process ontology* (in Publication)**
    - ➢ Query workpiece data: SPARQL SELECT
    - ➢ Generate sequence
    - ➢ Interconnect workpiece data (DSTV – LBD/ifcOWL)
    - ➢ Add resource (machine) including MQTT topic
    - ➢ Add tool including MQTT topic
    - ➢ Add workspace, constraints, actor, etc…
    - ➢ SPARQL UPDATE

- **Generate robot control data**
    - ➢ Query process data: SPARQL SELECT
    - ➢ Automated path-planning
    - ➢ Convert KRL code to linked data via *crc* ontology
    - ➢ SPARQL UPDATE

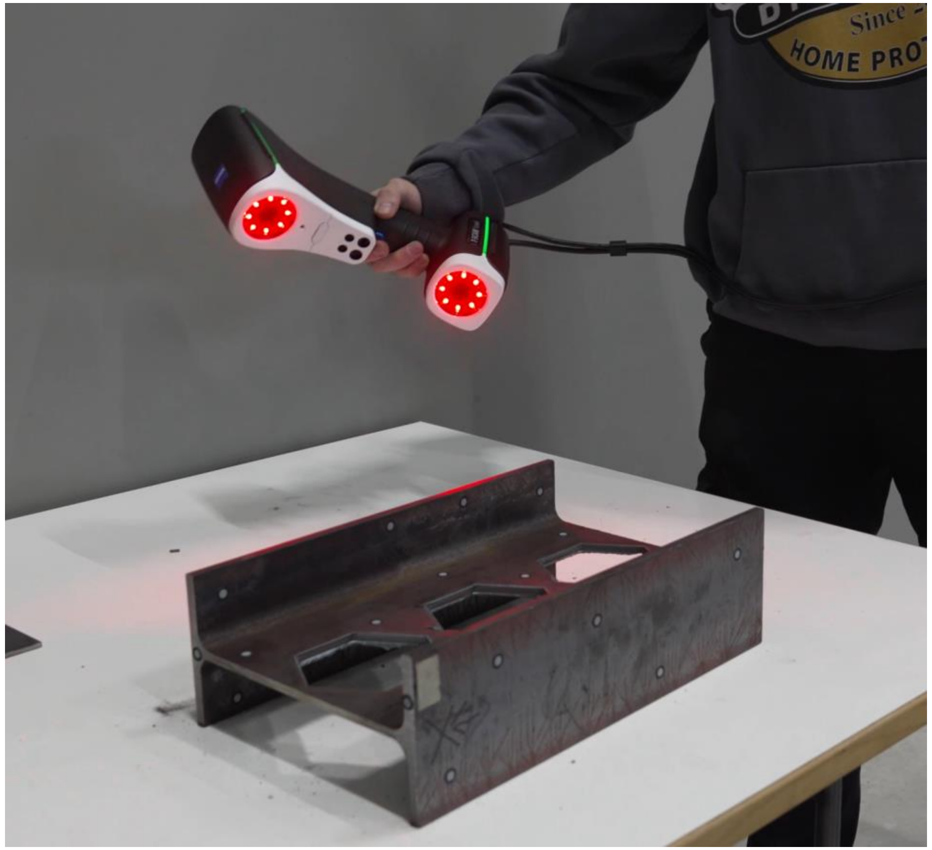


Sequence – Step 3: Plasma-Cutting

- **Running control script**
  - Query process: SPARQL UPDATE
  - Check constraints: SPARQL SELECT
  - Toggle Status: SPARQL UPDATE
  - Send commands via MQTT (Robot or Torch)
  - Wait …
  - Receive Feedback via MQTT
  - Update Status: SPARQL UPDATE
  - Repeat

Sequence – Step 4: Quality Control

- **Let the workpiece cool down**
  - Wait …
- **Compare to planning data**
  - Scan workpiece
  - Create mesh from pointcloud
  - Query planned geometry: SPARQL SELECT
  - *Best-Fit* Alogrithm to align
  - Measure Vertices
  - SPARQL UPDATE



Sequence – Step 5: Evaluation

- **Query process metadata**

DSTV data			ifcOWL	machine data				status feedback		
Name	**class**	**Feature**	**Output**	**cmdCount**	**cmd**	**dev**	**speed**	**TimeStart**	**TimeFinish**	**Duration**
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	PT 29.894S
first Cut	http://w3id.org/dstv#InternalContouring	http://baufest.org/dstv-test02#ContourList_0101	http://ip.rwth-aachen/baufest/ifc#IfcVoidingFeature_186	27	MoveLin	optimus	**5.0**	2023-03-28T15:43:55.770522+02:00	2023-03-28T15:45:03.253795+02:00	PT 1M7.483S
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	PT 28.372S
second Cut	http://w3id.org/dstv#InternalContouring	http://baufest.org/dstv-test02#ContourList_0201	http://ip.rwth-aachen/baufest/ifc#IfcVoidingFeature_170	27	MoveLin	optimus	**10.0**	2023-03-28T15:45:32.572661+02:00	2023-03-28T15:46:18.222345+02:00	PT 45.650S
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	PT 18.032S
third Cut	http://w3id.org/dstv#InternalContouring	http://baufest.org/dstv-test02#ContourList_0301	http://ip.rwth-aachen/baufest/ifc#IfcVoidingFeature_154	27	MoveLin	optimus	**20.0**	2023-03-28T15:46:37.193713+02:00	2023-03-28T15:47:00.235659+02:00	PT 23.042S
Travel to EndPOS	-	-	-	2	MoveAxis	optimus	-1.0	2023-03-28T15:47:00.643774+02:00	2023-03-28T15:47:10.531371+02:00	PT 9.888S

Sequence – Step 5: Evaluation

- **Calculate deviation & check tolerances**
  - Can be done directly via SPARQL SELECT & UPDATE

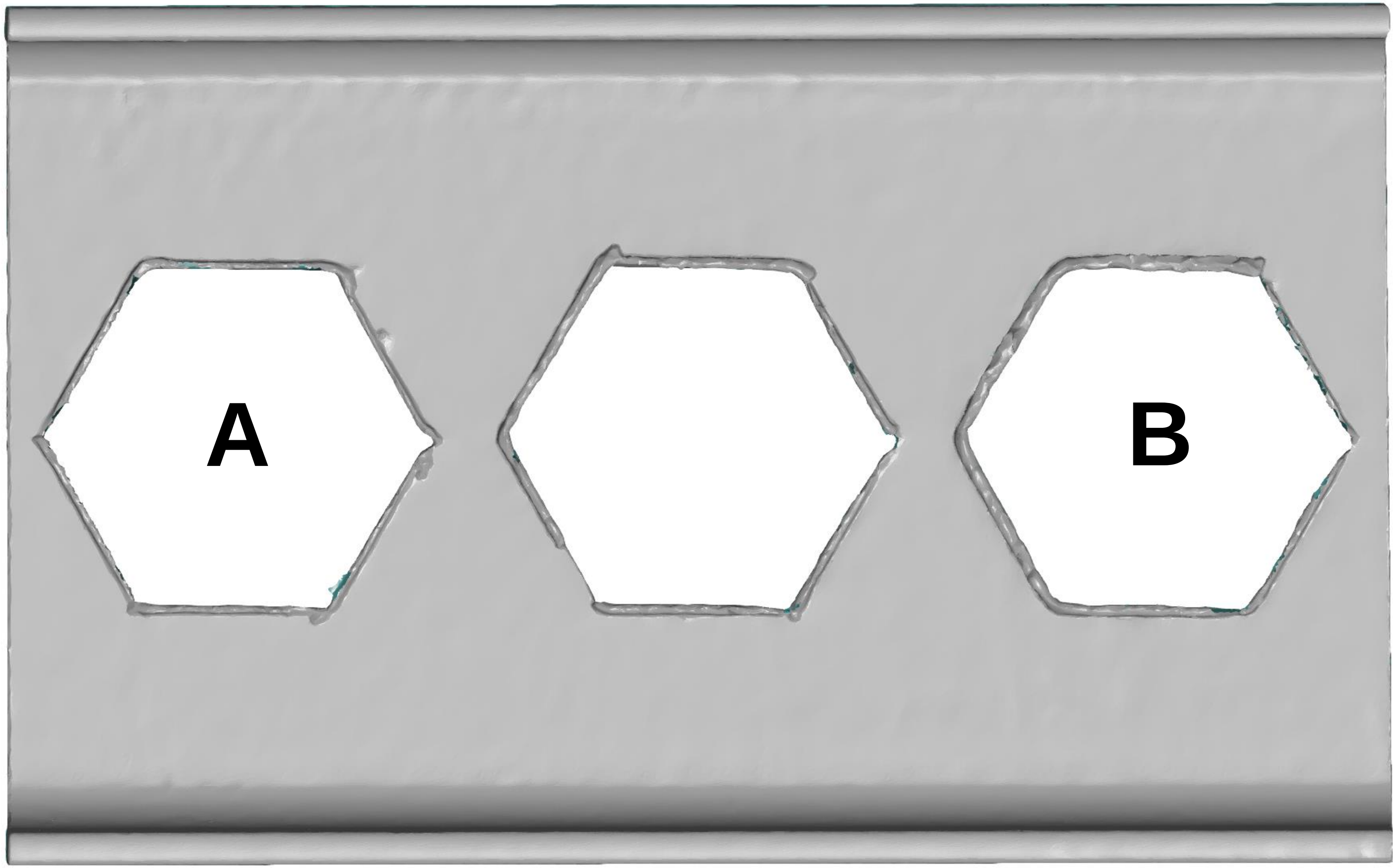
	planned			measured			deviation			
Feature	PlannedVertexX	PlannedVertexY	PlannedVertexZ	MeasuredVertexX	MeasuredVertexY	MeasuredVertexZ	VertexXDeviation	VertexYDeviation	VertexZDeviation	**tcpZ**
http://baufest.org/dstv-test02#Vertex_01_1	110.0	-10.5	**0.0**	109.858	-9.255	**0.304**	-0.141	1.244	**0.304**	**708.65**

- **Find patterns in the data**
  - Query deviations and machine data: SPARQL SELECT
  - Analyze / plot

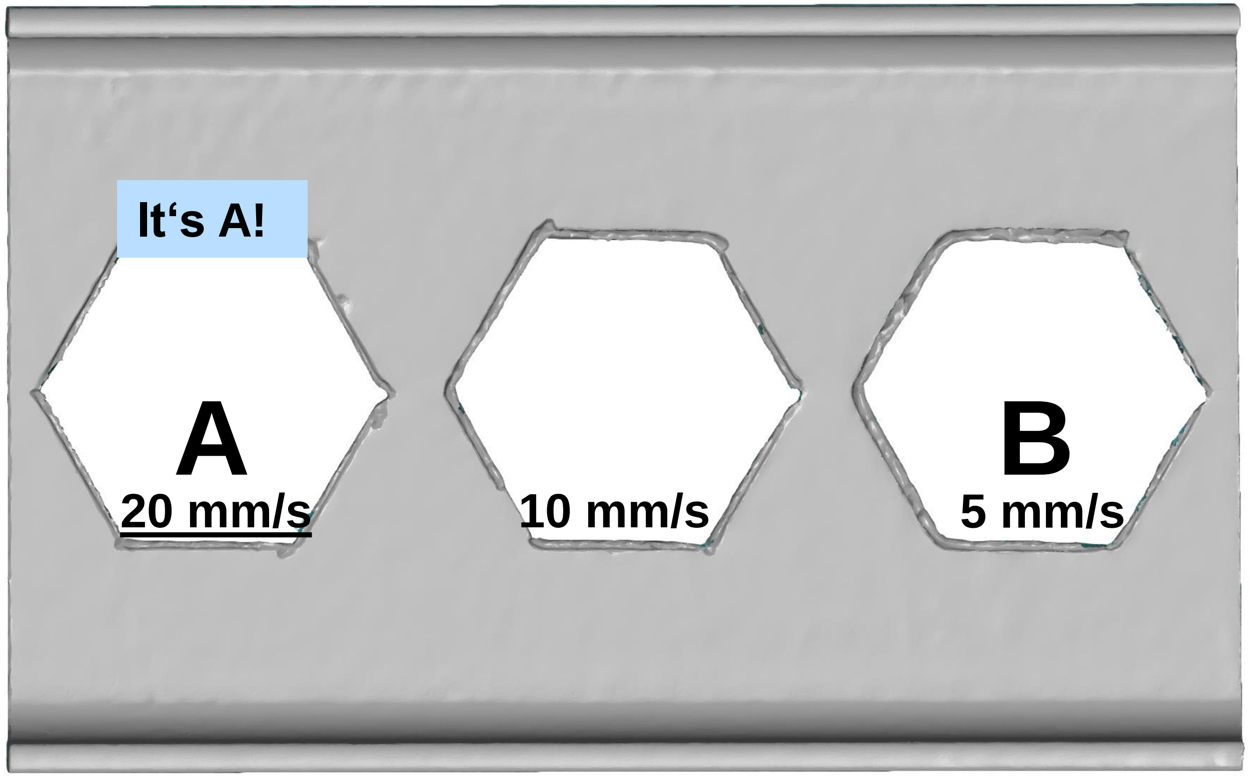
Name	cmd	dev	speed	tcpPose	Avg(AbsDevX)	Avg(AbsDevY)	Avg(AbsDevZ)	AbsTolXYZ	tcpZ
first Cut	MoveLin	optimus	5.0	http://baufest.org/dstv-test02#tcpPose_8c328310-a47c-5ac7-be84-d3db1d2007c3	0.368	0.547	4.079	-0.6 …+0.6	708.65
second Cut	MoveLin	optimus	10.0	http://baufest.org/dstv-test02#tcpPose_3050c77a-63b8-5869-a864-5635105b5023	0.231	0.504	3.399	-0.6 …+0.6	708.65
third Cut	MoveLin	optimus	20.0	http://baufest.org/dstv-test02#tcpPose_706a2a54-ba7f-5750-b8ed-6d99f1802774	0.190	0.413	3.086	-0.6 …+0.6	708.65

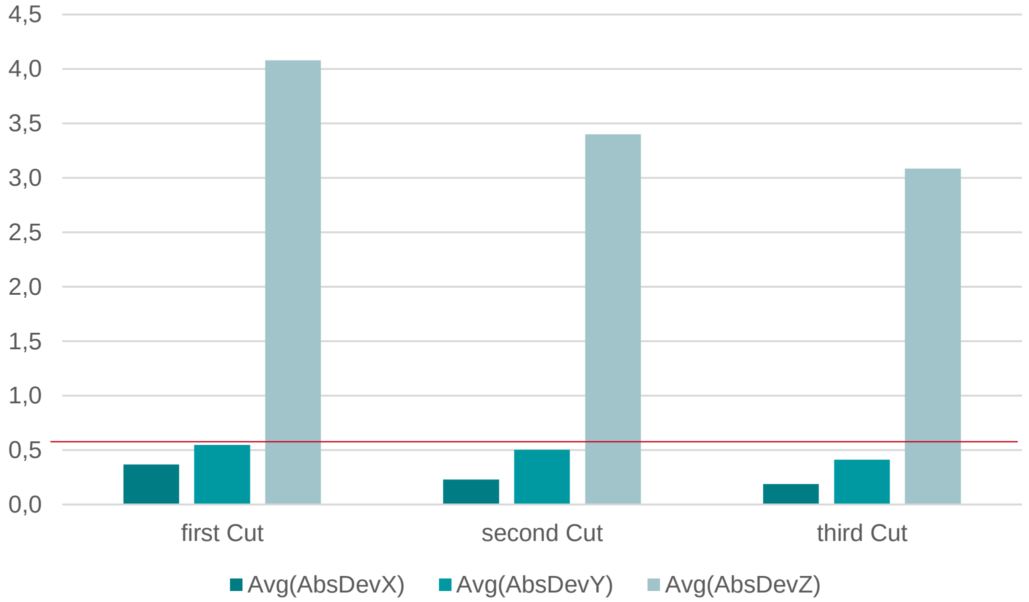
QUIZ

Which of the notches is the „third Cut“ produced with the highest speed ?



QUIZ





Name	cmd	dev	speed	tcpPose	Avg(AbsDevX)	Avg(AbsDevY)	Avg(AbsDevZ)	AbsTolXYZ	tcpZ
first Cut	MoveLin	optimus	**5.0**	http://baufest.org/dstv-test02#tcpPose_8c328310-a47c-5ac7-be84-d3db1d2007c3	0.368	0.547	4.079	-0.6 …+0.6	708.65
second Cut	MoveLin	optimus	**10.0**	http://baufest.org/dstv-test02#tcpPose_3050c77a-63b8-5869-a864-5635105b5023	0.231	0.504	3.399	-0.6 …+0.6	708.65
third Cut	MoveLin	optimus	**20.0**	http://baufest.org/dstv-test02#tcpPose_706a2a54-ba7f-5750-b8ed-6d99f1802774	0.190	0.413	3.086	-0.6 …+0.6	708.65

Conclusion & Outlook

Conclusion

- ✓ An autonomous robotic steel machining process using the proposed DSTV ontology can be realized
- ✓ Modeling the DSTV planning data via conversion of IFC to LBD/ifcOWL is easy to implement
- ✓ The extended concepts like measurement and tolerance data can be queried with rather simple SPARQL queries
- ✓ The proposed approach enables linking of production data with machine data and building element data
- ✓ Querying the data enables further optimization and can help to clarify process dependencies (~ *slow plasma cutting* ~ *bad results*)

- ➢ **Proposed approach is feasible**

Outlook

- ➢ Further extension of DSTV ontology needed (more processes)
- ➢ Dynamic status information: OPM vs RDF-Star?
- ➢ SHACL for process constraints?
- ➢ Further work on ontology-based capability matching
- ➢ Implementing the Asset Administration Shell – **AAS** for machine / network description
- ➢ Practical: Retrofit old CNC machine in the workshop ☺

Acknowledgments & References

Many thanks to our student assistants involved:

Timur Kuzu
Emre Ergin
Simon Laprell
Marica Vitt

The project *Bauplanungsorientiertes Fertigungsmanagement im Stahlbau 4.0* (BauFeSt 4.0 – project number 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 built upon results on of the research project *Internet of Construction* (IoC – project number 02P17D081).

References

[1] YouTube. (2019). *ZEMAN's "SBA2 Conti+" - The largest and most efficient Steel Beam Assembler*. *YouTube*. Retrieved June 6, 2023, from https://www.youtube.com/watch?v=MDjMuk4rWPA.

[2] Noy, N. & Mcguinness, Deborah. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Knowledge Systems Laboratory. 32.

[3] Rasmussen, Mads Holten & Lefrançois, Maxime & Bonduel, Mathias & Hviid, Christian & Karlshøj, Jan. (2018). OPM: An ontology for describing properties that evolve over time.