Towards A Visualisation Ontology for Reusable Visual Analytics

Baifan Zhou
baifanz@ifi.uio.no
SIRIUS Centre
University of Oslo, Norway

Zhipeng Tan
fixed-term.zhipeng.tan2@de.bosch.com
Bosch Center for AI, Germany

Zhuoxun Zheng
zhuoxun.zheng@de.bosch.com
Bosch Center for AI, Germany
Oslo Metropolitan University

Dongzhuoran Zhou
dongzhuoran.zhou@de.bosch.com
Bosch Center for AI, Germany
University of Oslo, Norway

Ognjen Savković
ognjen.savkovic@unibz.it
University of Bolzano, Italy

Evgeny Kharlamov
evgeny.kharlamov@de.bosch.com
Bosch Center for AI, Germany
University of Oslo, Norway

ABSTRACT

Data analytics including machine learning analytics is essential to extract insights from production data in modern industries. Visual analytics is essential for data analytics for e.g., presenting the data to provide an instinctive perception in exploratory data analysis, facilitating the presentation of data analysis results and the subsequent discussion on that. Visual analytics should allow a transparent common ground for discussion between experts in data analysis projects, given the multidisciplinary background of these experts. However, a standarised and formalised way of describing the knowledge and practice of visualisation is still lacking in the industry, which hamstrings the transparency and reusability of visual analytics. A visualisation ontology which models the nature and procedure of visualisation is well-suited to provide such standardisation. Currently a few studies discuss partially the modelling of visualisation, but insufficiently study the procedure of visualisation tasks, which is important for transparency and reusability especially in an industrial scenario. To this end, we present our ongoing work of development of the visualisation ontology in industrial scenarios at Bosch. We also demonstrate its benefits with case studies and knowledge graph based on our ontology.

1 INTRODUCTION

Data analytics including machine learning analytics [13, 14] aim to extract knowledge and insights from noisy, structured and unstructured data [3, 23, 33], and have been widely applied in industrial applications to reduce down-times, improve quality monitoring [15, 18, 38, 39], and robot positioning [2, 9]. Common data analytics include visual analytics, statistical analytics, machine learning analytics [20, 21, 37], etc. Among which, Visual analytics is arguably the most frequent activities because it is essential for various activities of data analytics [24, 30, 31] for e.g., presenting the data to provide an instinctive perception in exploratory data analysis [34, 40], facilitating the presentation of data analysis [28] results and subsequent discussion on that [16, 17].

Data analytics in the industry is of great importance [26, 27, 29], as the graphical presentation of the data helps to reach a common understanding and facilitates discussions among the stakeholders [6, 7, 32]. However, a standarised and formalised way of describing the knowledge [19, 41] and practice of visualisation is still lacking in the industry [36, 37], which hamstrings the transparency and reusability of visual analytics [22, 25]. A visualisation ontology which models nature and procedure of visualisation is well-suited to provide such standardisation, which, is formal explicit specifications of shared conceptualisations [4]. Furthermore, a visualisation ontology can provide many advantages: unambiguous definition of concepts that capture the domain knowledge of visual analytics, standardised description of visualisation procedures and modularised and resuable components [18, 35], construction of visual knowledge graph (KG) that represent executable visual analytical pipelines, etc.

Currently there are a few studies that discuss partially the modelling of visualisation, but they insufficiently depict the important elements in visual analytics and are less adequate in describing the practical procedure of visual analytics, which is important in the industrial scenarios for transparency, modularisation and reusability of visual analytical pipeline. For instance, computer science ontology [1, 12] contains general knowledge about visualisation, but the concepts of specific visualisation process is not involved. Statistics ontology [5, 11] enumerates the various visualisation methods, but insufficiently studies procedures of visualisation approaches.

To this end, we present our ongoing work of development of a visualisation ontology *VisuOnto*, with the industrial scenario of manufacturing data analytics at Bosch. We also demonstrate case studies and automated KG construction and verification of *VisuOnto*. In summary, our contributions are:

- We give detailed domain analysis of visual analytics (Section 3), including its common activities and procedure, from which we derive the three requirements for a visualisation ontology.
- We present a visualisation ontology (Section 4), *VisuOnto*, including the important classes, *Data*, *VisualMethod*, *VisualTask*, and *Constraints* that specify valid visual analytical pipeline.
- We demonstrate the evaluation and case studies of the *VisuOnto* (Section 5). We discuss two case studies that show *VisuOnto* can describe visual analytics in a transparent with visual KGs that represent the visual analytical pipeline in a modularised and

IJCKG'22, October 27-29, 2022, Hangzhou, China

ACM ISBN 978-1-4503-XXXX-X/22/12…$15.00
https://doi.org/XXXXXXX.XXXXXXX

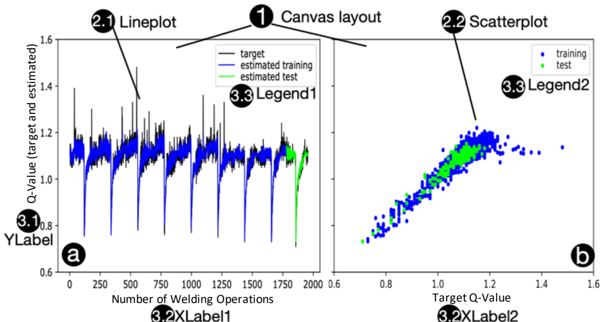

Figure 1: Example of machine learning visualisation and its procedure: ①create canvas layout; ②draw subplot; ③add description

reusable fashion. We further present the evaluation with competency questions and verification of the visual KGs.

2 ONTOLOGY DEVELOPMENT PROCESS

We broadly follow the methodology of *Ontology Development 101* [10]. We use Protege as ontology editor, and OWL 2 EL as the modelling language for its expressivity and polynomial time of query answering [8]. The modelling process can be divided into three steps:

Step 1: Domain Analysis. We discussed common visualisation activities with domain users (Bosch experts such as welding experts, data scientists) and read literature. We gather common and important terms in visualisation activities, including data, plot types, purposes, procedure, etc. In addition, we also studied frameworks and scripts of implementing visualisation with popular programming languages (Python which is the most popular among the users).

Step 2: Ontology Modelling. Based on the terms collected from Step 1, the concepts are formalised as classes. We classified them into categories to built the taxonomy of these classes, e.g., *CanvasTask* $\sqsubseteq$ *AtomicTask*, *VisualPipeline* $\sqsubseteq$ *VisualTask*. We formalise the relations between the classes as object properties, e.g., *VisualPipeline hasStartTask CanvasTask*.

Step 3: Evaluation and Case Study. We present several use cases that demonstrate the usage of *VisuOnto* and its evaluation in user study. Further more, we discuss the competence questions, and automated construction and verification of visual KGs.

3 DOMAIN ANALYSIS: VISUAL ANALYTICS

Visualisation Activities. We introduce the visual analytics that the ontology aims to cover. To limit the scope in this short paper and to have concrete discussion, we introduce the methodologies and concepts inspired by real-world cases of visual analytics in the industrial scenario of data analytics in manufacturing at Bosch. In particular, we study the following aspects of visualisation practice:

Data. The common data type include: vector, 2D matrix, and higher dimensional tensors. These data can be with or without interdependencies, e.g., a vector with temporal dependency becomes a sequence; a vector without temporal dependency is a single feature vector; a 2D matrix with spacial dependency becomes a image.

Plot Types. We study the visualisation types of line plot, scatter plot, bar chart, pie chart, histogram, heat map, etc. Although these methods can be applied to a broad range of data, there exist some heuristics that specify more suited methods for some types of data. E.g., line plot is more suited for data sequence; bar chart and pie chart are more suited for single feature vector with a limited length, because if the length is large, these two charts become too crowed.

Complexity Levels. We divide the visualisation plots into four complexity levels: (1) *simple figure*, which uses a single plot type to

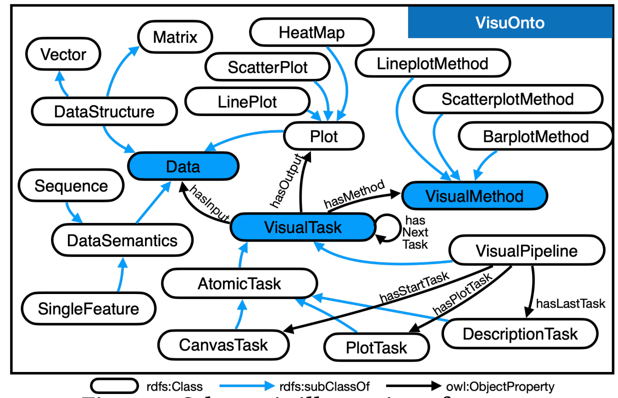

Figure 2: Schematic illustration of *VisuOnto*

visualise one data; (2) *multiple figure*, which repeatedly uses one plot type multiple times to visualise multiple data, e.g., Fig.1 displays two multiple figure; (3) *complex figure*, which uses multiple plot types to visualise multiple data (4) *multiple layout figure*, which contains multiple subplots that are of the previous three types, e.g., Fig.1 is a multiple layout figure with two multiple figures.

Purposes. We discuss two types purposes of visual analytics here: (1) *exploratory data analysis* to gain insights from the data, by dataset characterisation via visualisation of features, distribution, change, etc., outlier detection; (2) *machine learning visualisation* to provide an intuitive overview of the machine learning analysis results as a common ground for interpreting the machine learning models, e.g., Fig.1 visualises the prediction results of the quality indicator (Q-Value) in line plot and scatter plot.

Visualisation Procedure. In the aforementioned visualisation activities, the procedure to produce a figure can be largely divided into three steps: (1) create the canvas layout of the figure by giving the number and arrangement of subplots; (2) in each subplot, visualise the data by specifying the plot type (line plot, scatter plot, etc.) and plot properties (colour, line width, marker size, etc.); (3) add descriptions to each subplots, such as labels, legend, etc.

Three Aspects of Requirements. We now derive the following aspects of requirements for the visualisation ontology:

R1. Knowledge Capture: The ontology should be able to cover the aforementioned visualisation activities. Specifically, it should contain classes that describe the data, plot types, and their taxonomy. These classes should be able to support description of visual analytics of different complexity levels and purposes.

R2. Procedure: The ontology should allow description of the visualisation procedure. In particular, it should provide the schema for visual KGs that represent the procedure by a visual pipeline.

R3. Modularisation and Reusability: The ontology should enable description of modularised and resuable components of visual analytics. Specifically, the constructed visual KGs should allow modularised description of visual pipelines and they should be easily reused by modifying some components.

4 THE VISUAL ONTOLOGY

The *VisuOnto* (Fig.2) has 504 axioms, which define 30 classes, 11 object properties, and 142 datatype properties. It can be expressed in $\mathcal{EL}$ description logics. It contains three important upper classes: *Data*, *VisualMethod VisualTask*, which comprise a framework for describing visual analytics domain and serve as schema for visual KGs. Besides, *VisuOnto* contains a set of formal constrains that can be used for verifying correctness of visual KGs. Correctness means e.g.,

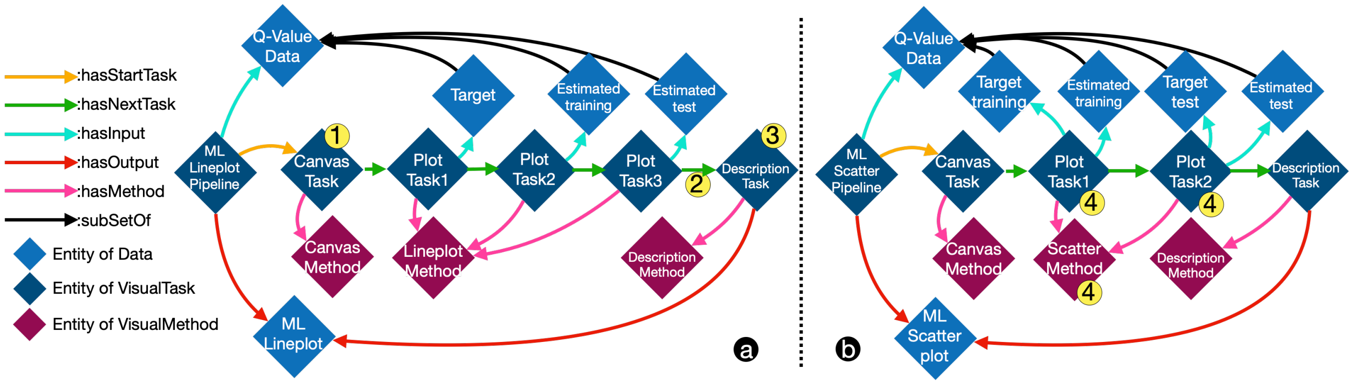

Figure 3: The visual KG in (a) for drawing Fig.1a can be easily reused for drawing Fig.1b by slightly modifying (a) to (b). These two can be further combined for drawing a multiple layout figure by modifying the entity *CanvasTask* ① and the *DescriptionTask* ③, and inserting the entities marked by ④ in (b) in the position marked by ② in (a).

visual analytical pipeline represented by visual KGs have necessary input data and correct methods for these data.

Data. The class *Data* is the upper class of all data classes in *VisuOnto*. It has three important sub-classes: *DataStruture*, *DataSemantics*, and *Plot*. The sub-classes of *DataStruture* describe the *dimension* of the data, including *Vector* that models 1D array, *Matrix* that models 2D matrix, and *Tensor* that models higher dimensional tensors. The sub-classes of *DataSemantics* describe the *meaning* of the data, e.g., *SingleFeature* models the feature that are independent, *Sequence* models the vectors that have order (temporal dependency, *MLResults* represent the data that are the analysis results of machine learning. We model the different plot types as sub-classes of *Plot*, these are usually the outputs of visual tasks, such as *LinePlot*, *ScatterPlot*, etc.

VisualMethod. The class *VisualMethod* is the upper class of all classes of visualisation methods in *VisuOnto*. Its sub-classes include e.g., *LineplotMethod*, *ScatterplotMethod*. Each method clearly has object properties connect to allowed data, e.g., $\exists AllowedData^{-}.LineplotMethod \sqsubseteq Array$. Each sub-class of *VisualMethod* represents a programming *script* of a visualisation function or module (several lines of code). The data type properties of the *VisualMethod* are the *arguments* of the *script*. E.g., *LineplotMethod* has data type properties such as *hasLineWidth*, *hasLineColour*, *hasLineStyle*.

VisualTask. The class *VisualTask* refers to the invoking of a visualisation method for solving a visualisation task. Each entity of *VisualTask* has inputs and is connected to entities of *Data* via *hasInput*. It also has method and is connected to an entity of *VisualMethod* via *hasMethod*. *VisualTask* has two sub-classes: (1) *AtomicTask* that represent indecomposable visualisation tasks, such as *PlotTask* that creates the visualisation of a single data, *CanvasTask* that specifies the canvas layout, and *DescriptionTask* that adds the labels, legends (see Visualisation Activities in Section 3); (2) *VisualPipeline*, consisting of a series of *AtomicTask*s, can create a figure with the canvas, drawings, and descriptions: An entity of *VisualPipeline* connected to an entity of *CanvasTask* via *hasStartTask*, which is connected to *PlotTask*s via *hasNextTask*, which is then connected to *DescriptionTask* via *hasNextTask* (Examples see Fig.3, Fig.4b).

Constraints. We formalise a set of constraints that should ensure the correctness of visual analytical pipelines represented by the visual KG. The correctness includes two aspects: (C1) the method and data should match. For example, the following axiom constraints any tasks which invoke plot method to have input data: $ax_1: \exists hasMethod.PlotMethod \sqsubseteq \exists hasInput$;(C2) the visual analytical pipelines should have the correct structure. For example, the following axiom constraints any plot method to have allowed data: $ax_2: PlotMethod \sqsubseteq \exists allowedData$.

5 EVALUATION

We evaluate our approach with an industrial scenario of visual analytics for automated welding at Bosch, which is an impactful welding process accounting for production of over 50 million cars globally every year. The sample dataset is collected from a factory in Germany, containing 22 welding machines and 53.2 million records.

5.1 Case Study with KGs

We present two examples for the two types of purposes of visual analytics introduced in Section 3.

Machine Learning Visualisation. The prediction results of Q-Values (an quality indicator of automated welding) in machine learning analysis are visualised in line plots in Fig. 1a, where the black lines indicate the target values to predict, the blue and green lines indicate the estimated values of the training and test sets, respectively. The line plots provide an intuitive way of understanding the prediction performance, as the perfect estimation should overlap with the target completely. Fig. 1b presents the scatter plots of the same results, with the x-axis as the target values and y-axis as the estimated values. The scatter plots provide another intuitive way of understanding the results, as the perfect estimation will lie in a straight line from bottom left corner to top right corner.

The generation of such figures can be described with the visual pipelines represented by the visual KGs in Fig. 3. The visual KG has an entity of *VisualPipeline*, with an entity of *CanvasTask* as the start task, four entities of *PlotTask* after it, and an entity of *DescriptionTask* as the last task. The *plot tasks* have input data, and all tasks are linked to entities of *VisualMethod* It can be seen that the visual knowledge provide a very *transparent* way of describing the *procedure* of visualisation (R2). The domain knowledge including various concepts (data, plot types) and activity of machine learning visualisation is well depicted by the ontology and KG (R1). Besides, Fig.1a can be easily reused for drawing Fig.1b by slightly modifying (a) to (b), and these two can be further combined for drawing a multiple layout figure by merging the entities of the two visual KGs, showing good modularisation and reusability (R3).

Exploratory Data Analysis. Here we present the example of inspecting dataset characteristics. The automated welding process produces a large volume of data. It is therefore very desired for welding experts to gain a quick overview of the data collected

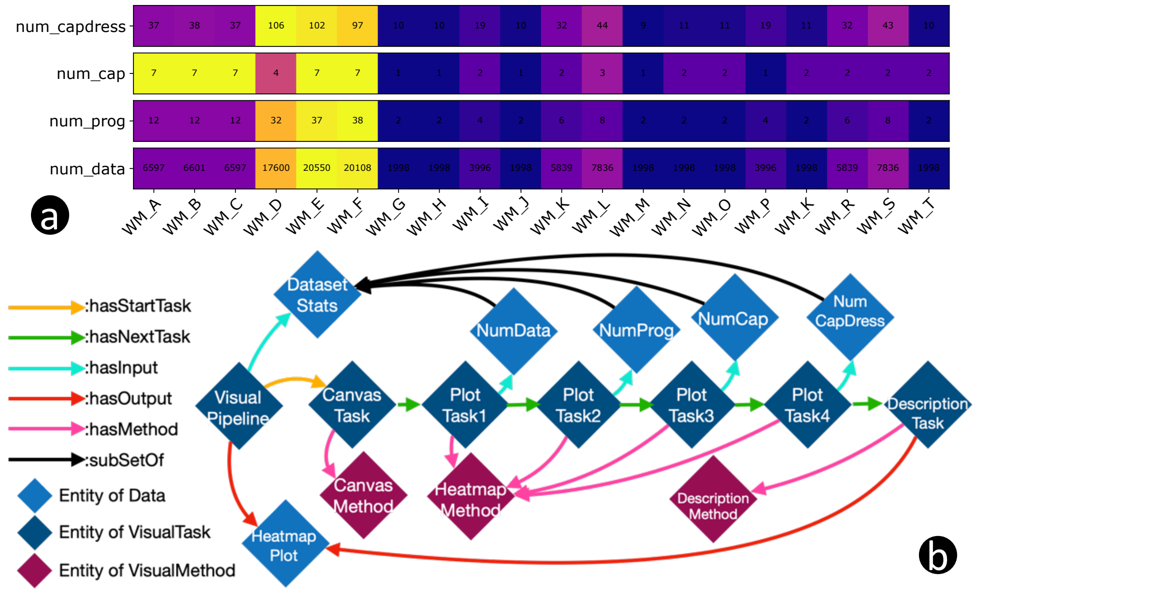

Figure 4: Example of overviewing dataset statistics in heat maps (a) and its visual KG (b). The colour map is a smooth continuum through yellow (large values indicating high data amount and complexity), purple (middle), and blue (small values with de-emphasised text).

from all welding machines, to learn the data amount, complexity produced by each machine in an intuitive way, and to narrow down the search realm for potential problematic welding machine (e.g., with excessive complexity due to anomalous maintenance).

Fig. 4a shows an overview of the general information of datasets in heat maps: The bottom row "num_data" summarises the number of data instances (each data instance is generated by one welding operation), providing an overview of the data amount; The row "num_prog" indicates the number of welding programs (a welding program prescribes the way of performing the welding operation), providing an overview of the data complexity since more welding programs indicate more complex dynamics in the data; The row "num_cap" and "num_capdress" provide further information of data complexity from the angle of maintenance, as cap change and cap dress are both maintenance operation of the welding process. We can see WM_A - WM_F have higher data complexity due to more cap changes, and WM_D - WM_F have higher data amount.

The generation of such heat maps can be described with a visual pipeline, represented by a visual KG in Fig. 4b. Similar to Fig.1, the visual pipeline has entities of *CanvasTask*, *PlotTask* and *DescriptionTask*, where the four entities of *PlotTask* have the same visual method, *HeatmapMethod*. This KG also provides a transparent way of procedure description (R2), captures the knowledge of heat map visual analytics. The tasks are well-described with four components that reuse the same method (R3).

Table 1: Competence Questions

#	Competence Question
C1	How many input data entities exist for the given visual task?
C2	What is the visual method chosen for visualisation?
C3	What type of plot does the pipeline generate?
C4	What data structure is allowed for line plot method?
C5	How many subplots exist in the visual pipeline?
C6	What colour is used for the give plot task?
C7	What is the first task in the visual pipeline?
C8	How many plot tasks exist in the visual pipeline?

5.2 Competency Questions

We evaluated *VisuOnto* with a set of competency questions (CQ), which are derived after discussion between welding experts, data scientists, and semantic experts. These CQs (examples given in Table 1) should reflect the coverage of the domain knowledge (R1) from three aspects: (I) actual/allowed input/output data or method for a visual task (C1 - C4), (II) datatype property for a visual task (C5 - C6), and (III) the workflow of a visual pipeline (C7 - C8). All of the CQs can be answered using SPARQL queries over KGs constructed with the *VisuOnto* as the schema.

5.3 Visual Analytical Pipeline Verification

The formally and explicitly expressed constraints in the *VisuOnto* (Section 4) enable the automatic verification of correctness of visual analytical pipelines. To do so, we rely on an OWL 2 reasoner and a set of SPARQL queries. The notion of correctness of the visual analytical represented by the visual KG is fulfilled, if and only if all the constraints explicitly contained within *VisuOnto* are satisfied. In particular, each constraint in *VisuOnto* is verified by a query, which is evaluated to false if the constraint is fulfilled. When we run all these queries over a visual KG that represent a visual analytical pipeline, if one of these queries is evaluated to be true, the visual analytical pipeline is verified to be incorrect. We take two queries Q_{ax_1} and Q_{ax_2} as examples, which correspond to the two constraint axioms ax_1 and ax_2, to verify whether the visualisation pipeline fulfils the two constraints:

$$Q_{ax_1} \leftarrow \mathit{VisualTask}(x) \land \mathit{hasMethod}(x, y) \land \mathit{PlotMethod}(y) \land \neg\exists z.\mathit{hasInput}(x, z)$$
$$Q_{ax_2} \leftarrow \mathit{PlotMethod}(x) \land \neg\exists y.\mathit{allowedData}(x, y)$$

6 CONCLUSION

In this paper we present our ongoing work of visualisation ontology *VisuOnto*, which encodes the knowledge and procedure of visual analytics in an ontology and knowledge graphs. Our evaluation with industrial use cases demonstrate the benefits of *VisuOnto* and it suffices the three requirements derived in the domain analysis.

7 ACKNOWLEDGMENTS

The work was partially supported by H2020 projects Dome 4.0 (Grant Agreement No. 953163), OntoCommons (Grant Agreement No. 958371), DataCloud (Grant Agreement No. 101016835) and SIRIUS Centre, Norwegian Research Council project number 237898.

REFERENCES

[1] Karin Koogan Breitman, Marco Antonio Casanova, and Walter Truszkowski. 2007. Ontology in computer science. *Semantic Web: Concepts, Technologies and Applications* (2007), 17–34.

[2] Onur Celik, Dongzhuoran Zhou, Ge Li, Philipp Becker, and Gerhard Neumann. 2022. Specializing Versatile Skill Libraries using Local Mixture of Experts. In *Conference on Robot Learning*. PMLR, 1423–1433.

[3] Vasant Dhar. 2013. Data science and prediction. *Commun. ACM* 56, 12 (2013), 64–73.

[4] Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What is an ontology? In *Handbook on ontologies*. Springer, 1–17.

[5] Konstantinos Kotis and Andreas Papasalouros. 2018. STATistics Ontology. http://stato-ontology.org/

[6] Dmitriy Mikhaylov, Baifan Zhou, Thomas Kiedrowski, Ralf Mikut, and Andrés-Fabián Lasagni. 2019. High Accuracy Beam Splitting Using SLM Combined With ML Algorithms. *Optics and Lasers in Engineering* 121 (2019), 227–235.

[7] Dmitriy Mikhaylov, Baifan Zhou, Thomas Kiedrowski, Ralf Mikut, and Andrés F Lasagni. 2019. Machine Learning Aided Phase Retrieval Algorithm for Beam Splitting With an LCoS-SLM. In *Laser Resonators, Microresonators, and Beam Control XXI*, Vol. 10904. International Society for Optics and Photonics, 109041M.

[8] Boris Motik et al. 2012. OWL 2 Web Ontology Language Profiles. https://www.w3.org/TR/owl2-profiles/

[9] Christoph Naab and Zhuoxun Zheng. 2022. Application of the unscented Kalman filter in position estimation a case study on a robot for precise positioning. *Robotics and Autonomous Systems* 147 (2022), 103904. https://www.sciencedirect.com/science/article/pii/S0921889021001895

[10] Natalya F Noy, Deborah L McGuinness, et al. 2001. Ontology development 101: A guide to creating your first ontology.

[11] Philippe Rocca-Serra and Susanna-Assunta Sansone. 2019. Experiment design driven FAIRification of omics data matrices, an exemplar. *Scientific Data* 6, 1 (2019), 1–4.

[12] Angelo A Salatino, Thiviyan Thanapalasingam, Andrea Mannocci, Francesco Osborne, and Enrico Motta. 2018. The computer science ontology: a large-scale taxonomy of research areas. In *International Semantic Web Conference*. Springer, 187–205.

[13] Rui Song, Anupama Hegde, Numan Senel, Alois Knoll, and Andreas Festag. 2022. Edge-Aided Sensor Data Sharing in Vehicular Communication Networks. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*. 1–7. https://doi.org/10.1109/VTC2022-Spring54318.2022.9860849

[14] Rui Song, Liguo Zhou, Venkatnarayanan Lakshminarasimhan, Andreas Festag, and Alois Knoll. 2022. Federated Learning Framework Coping with Hierarchical Heterogeneity in Cooperative ITS. *arXiv preprint arXiv:2204.00215* (2022).

[15] Yulia Svetashova, Baifan Zhou, Tim Pychynski, Stefan Schmidt, York Sure-Vetter, Ralf Mikut, and Evgeny Kharlamov. 2020. Ontology-enhanced machine learning: a Bosch use case of welding quality monitoring. In *International Semantic Web Conference*. Springer, 531–550.

[16] Yulia Svetashova, Baifan Zhou, Tim Pychynski, Stefan Schmidt, York Sure-Vetter, Ralf Mikut, and Evgeny Kharlamov. 2020. Ontology-Enhanced Machine Learning: A Bosch Use Case of Welding Quality Monitoring. In *ISWC*, Vol. 12507. Springer, 531–550.

[17] Yulia Svetashova, Baifan Zhou, Stefan Schmid, Tim Pychynski, and Evgeny Kharlamov. 2020. SemML: Reusable ML for Condition Monitoring in Discrete Manufacturing. In *ISWC (Demos/Industry)*, Vol. 2721. 213–218.

[18] Muhammad Yahya, Baifan Zhou, Zhuoxun Zheng, Dongzhuoran Zhou, John G Breslin, Muhammad Intizar Ali, and Evgeny Kharlamov. 2022. Towards generalized welding ontology in line with ISO and knowledge graph construction. *ESWC (Posters & Demos), Springer* (2022).

[19] Zhuoxun Zheng et al. 2022. Executable Knowledge Graph for Machine Learning: A Bosch Case for Welding Monitoring. In *ISWC*.

[20] Zhuoxun Zheng et al. 2022. Executable Knowledge Graph for Transparent Machine Learning in Welding Monitoring at Bosch. In *CIKM*.

[21] Zhuoxun Zheng et al. 2022. ExeKG: Executable Knowledge Graph System for User-friendly Data Analytics. In *CIKM*.

[22] Zhuoxun Zheng et al. 2022. Towards a Visualisation Ontology for Data Analysis in Industrial Applications. In *SemIIM at ESWC*.

[23] Zhuoxun Zheng, Baifan Zhou, Dongzhuoran Zhou, Gong Cheng, Ernesto Jiménez-Ruiz, Ahmet Soylu, and Evgeny Kharlamov. 2022. Query-based industrial analytics over knowledge graphs with ontology reshaping. *ESWC (Posters & Demos), Springer* (2022).

[24] Baifan Zhou. 2021. *Machine Learning Methods for Product Quality Monitoring in Electric Resistance Welding*. Ph. D. Dissertation. Karlsruhe Institute of Technology, Germany.

[25] Baifan Zhou et al. 2022. Knowledge Graph-Based Semantic System for Visual Analytics in Automatic Manufacturing. In *ISWC*.

[26] Baifan Zhou, Moncef Chioua, Margret Bauer, Jan-Christoph Schlake, and Nina F. Thornhill. 2019. Improving Root Cause Analysis by Detecting and Removing Transient Changes in Oscillatory Time Series With Application to a 1,3-Butadiene Process. *Industrial & Engineering Chemistry Research* 58 (2019), 11234–11250.

[27] Baifan Zhou, Moncef Chioua, and Jan-Christoph Schlake. 2017. Practical Methods for Detecting and Removing Transient Changes in Univariate Oscillatory Time Series. *IFAC-PapersOnLine* 50, 1 (2017), 7987–7992.

[28] Baifan Zhou, Tim Pychynski, Markus Reischl, Evgeny Kharlamov, and Ralf Mikut. 2022. Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding. *J. Intell. Manuf.* 33, 4 (2022), 1139–1163.

[29] Baifan Zhou, Tim Pychynski, Markus Reischl, and Ralf Mikut. 2018. Comparison of Machine Learning Approaches for Time-series-based Quality Monitoring of Resistance Spot Welding (RSW). *Archives of Data Science, Series A (Online First)* 5, 1 (2018), 13.

[30] Baifan Zhou, Damir Shakirov, Fabian Bleier, Juergen Haeufgloeckner, Martin Dieterle, Sean McConnell, Sinisa Slavnic, and Tim Pychynski. 2021. Method for Resistance Welding. US Patent App. 17/199,904.

[31] Baifan Zhou, Yulia Svetashova, Andre Gusmao, Ahmet Soylu, Gong Cheng, Ralf Mikut, Arild Waaler, and Evgeny Kharlamov. 2021. SemML: Facilitating development of ML models for condition monitoring with semantics. *J. Web Semant.* 71 (2021), 100664.

[32] Baifan Zhou, Yulia Svetashova, Tim Pychynski, and Evgeny Kharlamov. 2020. Semantic ML For Manufacturing Monitoring at Bosch. In *ISWC (Demos/Industry)*, Vol. 2721. 398.

[33] Baifan Zhou, Zhuoxun Zheng, Dongzhuoran Zhou, E Jimenez-Ruiz, G Cheng, T Tran, Daria Stepanova, Mohamed H Gad-Elrab, Nikolay Nikolov, Ahmet Soylu, et al. 2022. The data value quest: A holistic semantic approach at Bosch. *ESWC (Demos/Industry), Springer* (2022).

[34] Baifan Zhou, Zhuoxun Zheng, Dongzhuoran Zhou, Ernesto Jimenez-Ruiz, Gong Cheng, Trungkien Tran, Daria Stepanova, Mohamed H. Gad-Elrab, Nikolay Nikolov, Ahmet Soylu, and Evgeny Kharlamov. 2022. Exploiting the values of data: A holistic semantification approach at Bosch. In *ESWC (Demos/Industry)*. Springer.

[35] Baifan Zhou, Dongzhuoran Zhou, Jieying Chen, Yulia Svetashova, Gong Cheng, and Evgeny Kharlamov. 2021. Scaling Usability of ML Analytics With Knowledge Graphs: Exemplified with A Bosch Welding Case. In *IJCKG*.

[36] Dongzhuoran Zhou et al. 2022. ScheRe: Schema Reshaping for Enhancing Knowledge Graph Construction. In *CIKM*.

[37] Dongzhuoran Zhou et al. 2022. Towards Executable Knowledge Graph Translation. In *ISWC*.

[38] Dongzhuoran Zhou, Baifan Zhou, Jieying Chen, Gong Cheng, Egor Kostylev, and Evgeny Kharlamov. 2021. Towards Ontology Reshaping for KG Generation with User-in-the-Loop: Applied to Bosch Welding. In *The 10th International Joint Conference on Knowledge Graphs*. 145–150.

[39] Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Egor V Kostylev, Gong Cheng, Ernesto Jimenez-Ruiz, Ahmet Soylu, and Evgeny Kharlamov. 2022. Enhancing knowledge graph generation with ontology reshaping–Bosch case. *ESWC (Demos/Industry), Springer* (2022).

[40] Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Egor V. Kostylev, Gong Cheng, Ernesto Jimenez-Ruiz, Ahmet Soylu, and Evgeny Kharlamov. 2022. Ontology reshaping enhanced KG generation at Bosch. In *ESWC (Demos/Industry)*. Springer.

[41] Dongzhuoran Zhou, Baifan Zhou, Zhuoxun Zheng, Ahmet Soylu, Gong Cheng, Ernesto Jimenez-Ruiz, Egor V. Kostylev, and Evgeny Kharlamov. 2022. Ontology Reshaping for Knowledge Graph Construction: Applied on Bosch Welding Case. In *ISWC*. Springer.