

Towards Ontology Reshaping for KG Generation with User-in-the-Loop: Applied to Bosch Welding

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

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

Jieying Chen
jieyingc@ifi.uio.no
SIRIUS Centre, University of Oslo
Oslo, Norway

Gong Cheng
gcheng@nju.edu.cn
Nanjing University, China
Nanjing, China

Egor V. Kostylev
egork@ifi.uio.no
Department of Informatics
University of Oslo, NO

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

ABSTRACT

Knowledge graphs (KG) are used in a wide range of applications. The automation of KG generation is very desired due to the data volume and variety in industries. One important approach of KG generation is to map the raw data to a given KG schema, namely a domain ontology, and construct the entities and properties according to the ontology. However, the automatic generation of such ontology is demanding and existing solutions are often not satisfactory. An important challenge is a trade-off between two principles of ontology engineering: knowledge-orientation and data-orientation. The former one prescribes that an ontology should model the general knowledge of a domain, while the latter one emphasises on reflecting the data specificities to ensure good usability. We address this challenge by our method of ontology reshaping, which automates the process of converting a given domain ontology to a smaller ontology that serves as the KG schema. The domain ontology can be designed to be knowledge-oriented and the KG schema covers the data specificities. In addition, our approach allows the option of including user preferences in the loop. We demonstrate our on-going research on ontology reshaping and present an evaluation using real industrial data, with promising results.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; • **Computing methodologies** → **Knowledge representation and reasoning**; **Ontology engineering**.

KEYWORDS

ontology reshaping; knowledge graph generation; industrial application; user in the loop

*Dongzhuoran Zhou and Baifan Zhou contributed equally to this work as first authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

Bangkok '21, December 6–8, 2021, Online / Bangkok, Thailand

© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/10.1145/1122445.1122456>

ACM Reference Format:

Dongzhuoran Zhou, Baifan Zhou, Jieying Chen, Gong Cheng, Egor V. Kostylev, and Evgeny Kharlamov. 2021. Towards Ontology Reshaping for KG Generation with User-in-the-Loop: Applied to Bosch Welding. In *Bangkok '21: International Joint Conference on Knowledge Graphs, December 6–8, 2021, Online / Bangkok, Thailand*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Knowledge graphs (KGs) allow to structure information in terms of nodes and edges, where the nodes represent entities of interests, e.g., welding machines, and edges connect entities and thus represent relationships between them, e.g., by assigning software systems to concrete welding machines, or edges connect entities to their data values, e.g., by assigning the weight and price to welding machines [8]. In the context of Industry 4.0 [11] and Internet of Things [16], KGs have been successfully used in a wide range of applications and industrial sectors in well known production companies such as Bosch [34], Siemens [9], Festo [5], Equinor [21], etc.

An important challenge in scaling the use of KGs in industry is to facilitate automation of KG construction from industrial data due to its complexity and variety. Indeed, a common approach on KG construction is to construct entities and properties by relying on a given knowledge graph schema or ontology. The KG schema defines upper-level concepts of data, and consists of classes and properties. A classical domain ontology is a formal specification of shared conceptualisation of knowledge [7, 22] and it reflects experts knowledge on upper level concepts, specific domains, or applications.

However, since the domain ontologies are knowledge-oriented, they do not focus on the specificities of arbitrary datasets. In industry, data often come with a wide range of varieties. Many attributes exist in some datasets but not in others, and many terms in the domain ontologies do not exist in all datasets. Thus, directly using domain ontologies as KG schemata can naturally lead to a number of issues. Indeed, the resulting KG can contain a high number of blank-nodes, it may be affected by information loss, incomplete data coverage, and happen to be not user-friendly enough for applications.

Considering an example in automated welding [32, 33], which is an essential manufacturing process for producing the hundreds of thousands of car bodies in car factories everyday. The domain ontology (Figure 1a) shows that, in the welding process, the welding operations are operated under welding software systems, which

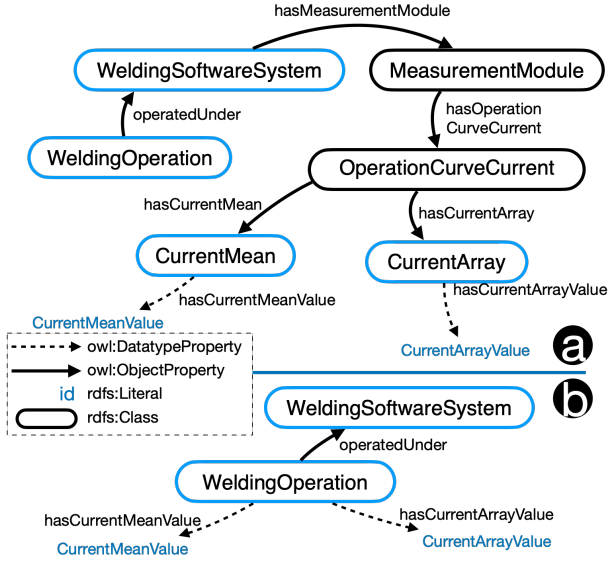


Figure 1: Why do we need ontology reshaping? Domain ontology (partially shown in a) reflects the knowledge; the KG schema (partially shown in b) needs to reflect raw data specificities and usability. Blue boxes: classes that can be mapped to attributes in the raw data; black boxes: classes that cannot be found in the raw data.

have measurement modules. These modules measure a sensor signal and save it as an operation curve, namely the current curve. The current curve is stored as an array, and it has a mean value. This presentation is created in close collaboration with the domain (welding) experts. It is thus intuitive for human to understand the domain knowledge well. Yet, the real datasets differ from the mental model. In most welding datasets, there exist one current mean value and one current array value for each welding operation (highlighted with blue boxes), while one welding software system is responsible for a huge group of welding operations, and there exist no attribute for the measurement module and operation curve current. If the domain ontology in Figure 1a is used for KG generation directly, the KG will contain a lot of blank nodes or dummy nodes generated by the class *MeasurementModule* and *OperationCurveCurrent*. Furthermore, the user will not be able to find the one-to-one correspondences between *WeldingOperation*, *CurrentMeanValue* and *CurrentArrayValue*. Meanwhile, the deep structure of the domain ontology will make the applications over the KG (e.g. query-based analytics) excessively complicated. Instead, the ideal schema for the KG generation would be Figure 1b, in which the ontology is significantly simplified. The generated KGs will have zero dummy nodes, and a much simpler structure. The KGs will be more efficient to generate, and easier to understand and use for the users.

The above-mentioned example exhibits the challenge of a trade-off between two principles of ontology engineering, (1) knowledge-orientation, which reflects the generality of domains, and (2) data-orientation, which aims at KG generation well-suited for specificities of data. The former principle focuses on conveying the meaning, or knowledge of a given domain, while the latter principle emphasises on the usability of ontologies for applications [10].

To address this challenge, we propose our *ontology reshaping* method that computes KG schemata from domain ontologies; these schemata in turn allow for generation of KGs of high quality. In this

way, we circumvent the trade-off problem and can use both ontologies for their different purposes. Our contributions are as follows:

- We propose an algorithm for ontology reshaping, which converts domain ontologies that reflect general knowledge to smaller KG schemata that cover data specificities, addressing the issue of sparse KG with dummy nodes.
- We design and conduct experiments for a proof-of-concept evaluation. We derive requirements for the use case, and design performance metrics for ontology reshaping.

2 PRELIMINARIES

Problem Formulation. Intuitively, ontology reshaping is about computing a smaller ontology from a larger one by taking a subset of its classes and possibly redefining some of its axioms based on some external heuristics as well as on some notion of optimality. In this work we do not aim at developing a formal theory of ontology reshaping, but rather at providing intuitions behind this problem and preliminary solutions that account for a particular type of reshaping. More precisely, in this work we consider reshaping where (i) re-definition of axioms is essentially “re-assigning” of properties from some classes to another (ii) the external heuristics is user’s input and (iii) the notion of optimality is the coverage of the data which the re-shaped ontology should be mapped to. In other words, such reshaping can be seen as:

$$S \leftarrow f(O, D, M, MC, I_U)$$

where f is an algorithm that takes in the inputs and outputs a KG schema S . The inputs are: O is a larger ontology, D is the raw data, that O is related to. The raw data D is in the form of relational table. M is a set of mappings that relate the attributes and table names in raw data to the classes in O . Apart from that, we need some more information given by the users. In our case, this includes two parts, (1) the users need to point out the most important entity (the most important class in O), named as the main class (MC); (2) User information I_U some more optional information.

Requirements of KG. We derive the requirements as follows:

- *R1 Completeness.* The knowledge graphs should be able to completely represent the raw data.
- *R2 Efficiency.* The generation of KG schema and KG should be efficient both in computational time and storage space.
- *R3 Simplicity.* The KG schema should not be over-complicated for the users to understand and use. In particular: (1) The generated KG should not have too much redundant information, e.g. dummy entities that are generated solely because of the schema but have no correspondence in the raw data; (2) the KG should not have complicated structures.

3 OUR METHOD: ONTOLOGY RESHAPING

Algorithm Explanation. Intuition of the algorithm in five steps:

- Step 1: Initialise the KG schema S with MC, and two sets from classes in O : potential classes C_E and potential properties C_P .
- Step 2: Add classes from C_E to S if they are mapped to table names T in M .
- Step 3: If properties in C_P are named as ‘ID’, ‘NAME’ etc, add their corresponding classes from C_E to S . Add further classes from C_E to S according to user information I_U (optional).

Algorithm 1: Ontology Reshaping

Input: O, M, MC, D, I_U
Output: S

- 1 $S \leftarrow \{MC\}, \{C_E, C_P\} \leftarrow O, A, T \leftarrow D$
- 2 $C_{ET} \leftarrow (M, T)$
- 3 $S := S \cup \{C_{ET}\}$
- 4 **foreach** $A_i \in A$ **do**
- 5 $C_{Pi} \leftarrow (M, A_i)$
- 6 **if** I_U **exists then**
- 7 $C_{Ei} \leftarrow \text{ClassIdentification}(C_{Pi}, I_U)$
- 8 $S := S \cup \{C_{Ei}\}$
- 9 **else if** 'ID' or 'Name' in C_{Pi} **then**
- 10 $C_{Ei} \leftarrow C_{Pi}$
- 11 $S := S \cup \{C_{Ei}\}$
- 12 $S \leftarrow \text{ClassConnection}(S, MC, O, I_U)$

- Step 4: Connect classes in S and connect properties to classes according to O .
- Step 5: Connect the rest classes in S to MC or according to user information I_U (optional)

Step 1. Initialisation. We start our algorithm with initialisation (Line 1). S is initialised with the main class MC . Then we map classes in O to attributes in D with M , where classes that can be mapped to attributes in M are initialised in the set of potential properties C_P , the rest classes are initialised as potential classes C_E .

Step 2. Class addition with table names. The raw data D is in form of relational table. If the relational table's name is able to be mapped from M to classes in C_E , We then add the mapped classes from C_E to S (Line 2 to 3).

Step 3. Entity identification by key words. In Step 3 we identify the classes in potential properties C_E by key words and (optional) user information. We map the attributes A stored in raw data from M to properties in C_P . Indeed, if properties in C_P are named as 'ID', 'NAME', e.g., `WeldingProgramID`, `WeldingMachineName`, we then add the corresponding classes in C_E , e.g., `WeldingProgram`, `WeldingMachine`, to S . Besides, we add the specified entities in S based on user information (Line 4 to 12).

Step 4. Classes connection . In Step 4 (in algorithm 2) we connect classes in S and connect properties to classes according to O . It takes 4 inputs: KG schema S , Domain ontology O , main class MC , and user information I_U . We start the Algorithm 2 with initialisation (Line 1), where the C_{rest} is initialised with copying classes in S . Then we iterate all permutation of the classes pairs (C_i, C_j) in S (Line 2). For each pair (C_i, C_j) , if the relation $r_o(C_i, C_j)$ already exists in S , we then continue to the next pair (Line 3 to 4). Otherwise, if direct relation $r_{od}(C_i, C_j)$ exists in O but not in S (Line 5 to 6), then this relation will be added to S . In case of existing in the ontology an indirect relation between the two classes C_i, C_j , the algorithm first evaluates if user information I_U exists and adds a relation between classes $r_o(C_i, C_j)$ (Line 7 to 10), and if not both classes are related to the main class $r_o(MC, C_i), r_o(MC, C_j)$ (Line 11 to 12).

Algorithm 2: Class Connection

Input: S, MC, O, I_U
Output: S

- 1 $C_{rest} \leftarrow \text{ClassExtraction}(S)$
- 2 **foreach** $(C_i, C_j) \subseteq S$ **do**
- 3 **if** $r_o(C_i, C_j) \in S$ **then**
- 4 Continue
- 5 **else if** $r_o(C_i, C_j) \in O$ **then**
- 6 $S := S \cup \{r_o(C_i, C_j)\}$
- 7 **else if** $r_{oind}(C_i, C_j) \in O$ **then**
- 8 **if** I_U **exists then**
- 9 $\{r_o\} \leftarrow \text{UserInfoExtraction}(I_U)$
- 10 $S := S \cup \{r_o(C_i, C_j)\}$
- 11 **else**
- 12 $S := S \cup \{r_o(MC, C_i), r_o(MC, C_j)\}$
- 13 $C_S \leftarrow \text{ClassExtraction}(S)$
- 14 $C_{rest} \leftarrow C_{rest} \setminus C_S$
- 15 **foreach** $C_i \in C_{rest}$ **do**
- 16 **if** I_U **exists then**
- 17 $\{r_d(C_i, C_u), r_d(C_u, C_i)\} \leftarrow \text{UserInfoExtraction}(I_U)$
- 18 $S := S \cup \{r_d(C_i, C_u), r_d(C_u, C_i)\}$
- 19 **else**
- 20 $S := S \cup \{r_d(MC, C_i)\}$

Step 5. Classes connection with UserInfo. In Step 5 (in algorithm 2) we connect classes in S to MC that are not connected to any classes, or according to `UserInfo` I_U (optional) (Line 18 to 25).

4 EVALUATION

Data Description. We have five inputs: domain ontology O , dataset D , mappings M , main class MC and user information `UserInfo`.

The domain ontology O is an OWL 2 ontology and can be expressed in Description Logics $\mathcal{SHI}(\mathcal{D})$. With its 1249 axioms, which contain 147 classes, 145 object properties, and 132 datatype properties, it models the general knowledge of a type of fully automated welding process, resistance spot welding [25, 34, 38].

The industrial dataset D is collected from welding production lines in a factory in Germany. The raw form of D is various, including txt, csv, RUI (a special format of Bosch time series data), SQL database, etc. They are transformed into relational tables. We selected a subset from the huge amount of data to evaluate our method. The transformed data contain one huge table of welding operation records and a series of tables of welding sensor measurements. In total, there exist about 4.315 million records. These data account for 1000 welding operations, estimated to be related to 100 cars.

The mapping M consists of three mappings: (1) a meta mapping that gives the correspondence between the table names in D and the class names in O ; (2) an operation mapping that relates the attribute names of the welding operation records to the classes in O ; (3) a sensor mapping that annotates the attribute names of the welding sensor measurements with the classes in O .

The `UserInfo` contains two parts, (1) mandatory: the users need to point out the main entity ME , which is the welding operation; (2)

optional: other information, e.g. the attribute names corresponding to other possible entity names and their related properties

Experiment Design. To test whether our ontology reshaping algorithm can perform well for an arbitrary dataset, we randomly sub-sample the dataset D to 6 sub-datasets (Set 1-6 in Table 1). Each set contains a subset of the attributes of D , reflecting different data complexity. The numbers of attributes in the subsets increase by ten each time, from 10 to 60. We repeat the sub-sampling for each subset 10 times to decrease the randomness, and provide the mean values of the evaluation metrics (introduced in next paragraph).

We compare two approaches: the *baseline* of KG generation without ontology reshaping; KG generation with our method of ontology reshaping. The baseline is a naive approach to select a subset from the domain ontology O and use it as the KG schema. The subset includes (1) classes that have correspondence to attributes in raw data, and (2) the classes connect these classes in (1).

In total, we performed 120 experiments ($2 \times 6 \times 10$): 2 approaches (the baseline and KG generation with ontology reshaping), 6 times sub-sampling with 10 repetition for each.

Evaluation Metrics. We use three sets of metrics to evaluate the fulfilment of the three requirements in Sect. 2: completeness, efficiency and simplicity. To evaluate *completeness*, we have *data coverage* that represents the percentage of attributes in raw data covered by the generated KG. *Efficiency* measures the ability of the approaches to use the least time cost for generating KG from raw data and to take the least storage space and the least entities and properties to represent the same information of raw data. The efficiency metrics thus include time cost, storage space, number of classes in the KG schema, and number of entities and properties (including object properties and data properties) in the generated KG. *Simplicity* measures the performance of the approaches to represent the same raw data with the simplest KG. These metrics include the number of dummy entities in the generated KG (namely the entities that cannot be mapped to attributes in the raw data, but are generated solely because of the KG schema), and two depths (the depth characterise the number of edges to connect two nodes in the KG via the shortest path): (1) root to leaf depth measures the “depth” to find the furthest entity starting from the main entity; (2) the global depth is the largest depth across the whole KG.

Results and Discussion. In Table 1, the six subsets with different number of randomly sub-sampled (each repeated 10 times) attributes are listed as the columns. The data complexity increases from Set 1 to Set 6. Both approaches can represent the raw data 100%. Thus it is not displayed in the table. We observe that OntoReshape outperforms the baseline in terms of time cost, storage space and other efficiency metrics, especially that the KG generation with OntoReshape is 7 to 8 times fast than that of the baseline. In terms of simplicity, OntoReshape also outperforms the baseline significantly. The baseline generates a huge amount of dummy entities, while OntoReshape generates zero dummy entities, drastically reducing information redundancy. The two depths of the KGs generated by OntoReshape are also only half or one third of that of the baseline. The KGs generated by OntoReshape are thus much more simpler, and easier to understand for the users. With an average root to leaf depth of only 1.2, we can see the users can use fewer queries to reach the deepest entities in the KG generated by OntoReshape.

Subset		Set 1	Set 2	Set 3	Set 4	Set 5	Set 6
Raw data	#attributes	10	20	30	40	50	60
	storage space (kB)	81.3	164.7	222.5	309.2	379.5	451.3
Efficiency metrics							
Baseline	time cost (sec)	139.3	298.2	429.5	551.6	723.3	836.9
	storage space (MB)	1.3	2.0	2.7	3.4	4.1	4.8
	#avg. class	147.0	147.0	147.0	147.0	147.0	147.0
	#max. class	147.0	147.0	147.0	147.0	147.0	147.0
	#object prop.	14.1k	25.0k	34.8k	44.6k	54.3k	64.3k
	#data prop.	404.5	577.8	1174.4	1546.9	2219.1	2100.4
	#entities	1704.5	1977.8	2474.4	3346.9	4319.1	4048.4
Onto-Reshape	time cost (sec)	21.6	40.8	61.2	84.9	113.3	128.0
	storage space (MB)	0.5	0.8	1.2	1.6	2.0	2.4
	#avg. class	1.3	1.4	1.3	1.8	2.1	1.9
	#max. class	2.0	2.0	3.0	3.0	3.0	3.0
	#avg. object prop.	300.0	400.0	306.0	806.8	1120.4	913.6
	#avg. data prop.	7.9k	14.6k	22.9k	29.7k	37.5k	45.3k
	#entities	1013.6	1020.4	1014.6	1028.2	1050.6	1029.2
Simplicity metrics							
Baseline	#avg. dummy entities	4100.0	4600.0	4800.0	4800.0	4600.0	4800.0
	#max. dummy entities	5000.0	5000.0	6000.0	6000.0	6000.0	6000.0
	avg. root to leaf depth	3.2	3.3	3.3	3.3	3.4	3.3
	max. root to leaf depth	4.0	4.0	4.0	4.0	4.0	4.0
	avg. global depth	3.2	3.3	3.3	3.3	3.4	3.3
	max. global depth	4.0	4.0	4.0	4.0	4.0	4.0
Onto-Reshape	#avg. dummy entities	0.0	0.0	0.0	0.0	0.0	0.0
	#max. dummy entities	0.0	0.0	0.0	0.0	0.0	0.0
	avg. root to leaf depth	1.3	1.4	1.2	1.7	2.1	1.7
	max. root to leaf depth	2.0	2.0	2.0	2.0	3.0	2.0
	avg. global depth	1.3	1.4	1.3	1.8	2.1	1.7
	max. global depth	2.0	2.0	3.0	3.0	3.0	2.0

Table 1: Evaluation on subsets with different numbers of attributes shows that KG generation with the ontology reshaping (Onto-Reshape) outperforms the baseline significantly in terms of efficiency and simplicity. The data coverage of both methods is 100%, and is thus not displayed in the table. Avg.: average, prop.: property.

5 RELATED WORK

Knowledge graphs have received much attention in industries [6, 9, 20, 28]. KGs provide semantically structured information that can be interpreted by computing machines [27, 39], and an efficient foundation for standardised ways of data retrieval and analytics to support data driven methods. Data driven methods have been widely used in industries [17, 18, 30, 31], especially machine learning [26, 29, 35–37]. The problem of transforming a bigger ontology to a smaller ontology of the same domain is often referred to as ontology modularisation [1–4, 14] and ontology summarisation [19]. Most of them focus on the problem of selecting a subset of the ontology that is interesting for the users [12], but they still cannot avoid dummy entities. Works on ontology reengineering [23, 24] also talked about reuse/adjustment of ontologies, but they do not focus on automatically creating an ontology that reflect data specificities.

6 CONCLUSION AND OUTLOOK

This work presents our on-going research of ontology reshaping, that generates a small ontology covering data specificities from a more complex domain ontology reflecting general knowledge. As future work, we plan to study the compatibility between the domain ontologies and KG schemata, i.e. to ensure that the semantics of the domain is respected in the smaller ontology. The current approach can not fully retain the semantics of the domain ontology. This can be addressed by uniform interpolation, also known as forgetting [13, 15], which we also plan to study in the future.

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

REFERENCES

- [1] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. 2017. Zooming in on Ontologies: Minimal Modules and Best Excerpts. In *Proc. of ISWC'17*. Springer, 173–189.
- [2] Jieying Chen, Michel Ludwig, Yue Ma, and Dirk Walther. 2019. Computing Minimal Projection Modules for $\mathcal{E}\uparrow^{\mathcal{R}}$ -Terminologies. In *Proc. of JELIA'19*, Vol. 11468. Springer, 355–370.
- [3] Jieying Chen, Michel Ludwig, and Dirk Walther. 2016. On Computing Minimal $\mathcal{E}\uparrow$ -Subsumption Modules. In *Proc. of WOMoCoE'16*. CEUR-WS.org.
- [4] Jieying Chen, Michel Ludwig, and Dirk Walther. 2018. Computing Minimal Subsumption Modules of Ontologies. In *Proc. of GCAI'18*. EasyChair, 41–53.
- [5] Stefan Elmer, Foued Jrad, Thorsten Liebig, Anees ul Mehdi, Michael Opitz, Thomas Stauß, and Dirk Weidig. 2017. Ontologies and Reasoning to Capture Product Complexity in Automation Industry. In *ISWC Posters & Demonstrations and Industry Tracks*.
- [6] Martina Garofalo, Maria Angela Pellegrino, Abdulrahman Altabba, and Michael Cochez. 2018. Leveraging Knowledge Graph Embedding Techniques for Industry 4.0 Use Cases. In *Cyber Defence in Industry 4.0 Systems and Related Logistics and IT Infrastructures*. IOS Press, 10–26.
- [7] Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What Is an Ontology? In *Handbook on ontologies*. Springer, 1–17.
- [8] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge Graphs. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–37.
- [9] Thomas Hubauer, Steffen Lamparter, Peter Haase, and Daniel Markus Herzog. 2018. Use Cases of the Industrial Knowledge Graph at Siemens. In *International Semantic Web Conference (P&D/Industry/BlueSky)*.
- [10] Mustafa Jarrar. 2005. Towards Methodological Principles for Ontology Engineering. Available at SSRN 1117002 (2005).
- [11] Henning Kagermann. 2015. Change Through Digitization—Value Creation in the Age of Industry 4.0. In *Management of permanent change*. Springer, 23–45.
- [12] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. 2013. Model-Theoretic Inseparability and Modularity of Description Logic Ontologies. *Artificial Intelligence* 203 (2013), 66–103.
- [13] Boris Konev, Dirk Walther, and Frank Wolter. 2009. Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, Craig Boutilier (Ed.), 830–835. <http://ijcai.org/Proceedings/09/Papers/142.pdf>
- [14] Patrick Koopmann and Jieying Chen. 2020. Deductive Module Extraction for Expressive Description Logics. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 1636–1643.
- [15] Michel Ludwig and Dirk Walther. 2016. Towards a Practical Decision Procedure for Uniform Interpolants of EL-Tboxes - A Proof-Theoretic Approach. In *GCAI 2016. 2nd Global Conference on Artificial Intelligence, September 19 - October 2, 2016, Berlin, Germany (EPIC Series in Computing, Vol. 41)*, Christoph Benzmüller, Geoff Sutcliffe, and Raúl Rojas (Eds.). EasyChair, 147–160. <https://doi.org/10.29007/bsqm>
- [16] Adrian McEwen and Hakim Cassimally. 2013. *Designing the Internet of Things*. John Wiley & Sons.
- [17] Dmitry Mikhaylov, Baifan Zhou, Thomas Kiedrowski, Ralf Mikut, and Andrés-Fabíán Lasagni. 2019. High Accuracy Beam Splitting Using SLM Combined With ML Algorithms. *Optics and Lasers in Engineering* 121 (2019), 227–235.
- [18] Dmitry Mikhaylov, Baifan Zhou, Thomas Kiedrowski, Ralf Mikut, and Andrés F Lasagni. 2019. ML 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, San Francisco, USA, 109041M.
- [19] Tugba Ozacar and Ovuuc Ozturk. 2021. Karyon: A Scalable and Easy to Integrate Ontology Summarisation Framework. *Journal of Information Science* 47, 2 (2021), 255–268.
- [20] Martin Ringsquandl, Evgeny Kharlamov, Daria Stepanova, Steffen Lamparter, Raffaello Lepratti, Ian Horrocks, and Peer Kröger. 2017. On Event-Driven Knowledge Graph Completion in Digital Factories. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 1676–1681.
- [21] Martin G. Skjæveland, Espen H. Lian, and Ian Horrocks. 2013. Publishing the Norwegian Petroleum Directorate's Factpages as Semantic Web Data. In *ISWC*. 162–177.
- [22] Barry Smith. 2012. Ontology. In *The furniture of the world*. Brill, 47–68.
- [23] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. 2012. The Neon Methodology for Ontology Engineering. In *Ontology engineering in a networked world*. Springer, 9–34.
- [24] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, and Aldo Gangemi. 2012. Introduction: Ontology Engineering in a Networked World. In *Ontology engineering in a networked world*. Springer, 1–6.
- [25] 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*.
- [26] Yulia Svetashova, Baifan Zhou, Stefan Schmid, Tim Pychynski, and Evgeny Kharlamov. 2020. SemML: Reusable ML Models for Condition Monitoring in Discrete Manufacturing. In *ISWC (Demos/Industry)*, Vol. 2721. 213–218.
- [27] Zhanfang Zhao, Sung-Kook Han, and In-Mi So. 2018. Architecture of Knowledge Graph Construction Techniques. *International Journal of Pure and Applied Mathematics* 118, 19 (2018), 1869–1883.
- [28] Pai Zheng, Liqiao Xia, Chengxi Li, and Bufan Liu. 2021. Towards Self-X Cognitive Manufacturing Network: An Industrial Knowledge Graph-Based Multi-Agent Reinforcement Learning Approach. *Journal of Manufacturing Systems* 61 (2021), 16–26.
- [29] Baifan Zhou. 2021. *Machine Learning Methods for Product Quality Monitoring in Electric Resistance Welding*. Ph.D. Dissertation. Karlsruhe Institute of Technology, Germany.
- [30] 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, 26 (2019), 11234–11250.
- [31] 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.
- [32] 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)* (2018).
- [33] Baifan Zhou, Damir Shakirov, Fabian Bleier, Juergen Haeufgloekner, Martin Dieterle, Sean McConnell, Sinisa Slavnic, and Tim Pychynski. 2021. Method for Resistance Welding. US Patent App. 17/199,904.
- [34] Baifan Zhou, Yulia Svetashova, Seongsu Byeon, Tim Pychynski, Ralf Mikut, and Evgeny Kharlamov. 2020. Predicting Quality of Automated Welding With Machine Learning and Semantics: A Bosch Case Study. In *CIKM*.
- [35] 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. *Journal of Web Semantics* 71 (2021), 100664.
- [36] Baifan Zhou, Yulia Svetashova, Tim Pychynski, and Evgeny Kharlamov. 2020. Semantic ML For Manufacturing Monitoring at Bosch. In *ISWC (Demos/Industry)*, Vol. 2721. 398.
- [37] Baifan Zhou, Yulia Svetashova, Tim Pychynski, and Evgeny Kharlamov. 2020. SemFE: Facilitating ML Pipeline Development With Semantics. In *CIKM*.
- [38] Baifan Zhou, Dongzhuoran Zhou, Jieying Chen, Yulia Svetashova, Gong Cheng, and Evgeny Kharlamov. 2021. Scaling ML Analytics With Knowledge Graphs: A Bosch Welding Case. In *IJCKG*.
- [39] Xiaohan Zou. 2020. A Survey on Application of Knowledge Graph. In *Journal of Physics: Conference Series*, Vol. 1487. IOP Publishing, 012016.