Towards a Visualisation Ontology for Data Analysis in Industrial Applications

Zhuoxun Zheng[1,2,*], Baifan Zhou[3], Ahmet Soylu[2,3,4] and Evgeny Kharlamov[1,3]

[1]Bosch Center for Artificial Intelligence, Renningen, Germany
[2]Department of Computer Science, Oslo Metropolitan University, Oslo, Norway
[3]SIRIUS Centre, University of Oslo, Oslo, Norway
[4]Department of Computer Science, Norwegian University of Science and Technology

Abstract

Machine learning (ML) approaches have proven their great potential in dealing with heterogeneous and voluminous data and thus are widespread in industry. To facilitate the presentation of the ML results and the subsequent discussion on that, visualisation is essential, as it effectively conveys the information behind the data. However, a standardisation of the knowledge and practice about visualisation is still lacking in the industry, which sometimes leads to misunderstandings in conveying information and thus making the discussions on the ML results error-prone. A visualisation ontology which models the nature and pipeline of visualization tasks are well suited to provide such standardisation. Currently there are a few studies that discuss partially the modelling of visualisation, however they are less adequate in depicting the practical procedure of visualisation tasks, which is highly demanded in the industrial applications. To this end, we present our ongoing work of development of the visualisation ontology in industrial applications at Bosch. We also discuss applications and evaluation of our ontology.

Keywords

Ontology Engineering, Knowledge Graph, KG Generation, Data Science, Manufacturing

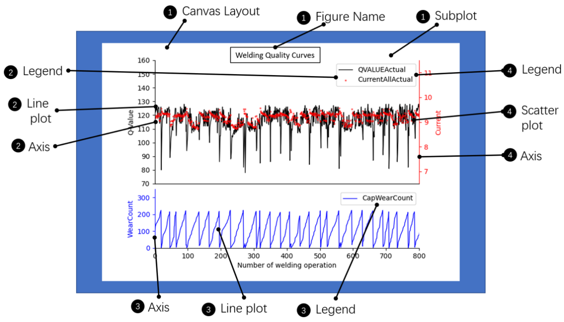
1. Introduction

Data driven methods especially machine learning aim to extract knowledge and insights from noisy, structured and unstructured data [1, 2, 3], and have been widely applied in industrial applications to reduce down-times, improve quality monitoring [4, 5, 6, 7], and robot positioning [8, 9]. Machine learning approaches have proven their great potential in dealing with heterogeneous and voluminous data, which is common in the industry, and thus greatly contributes to the overall value-chain [10]. After the machine learning approaches, the visualisation of the results is also of great importance, as the graphical presentation of the results helps to reach a common understanding and facilitates subsequent discussions among the stakeholders.

However, a formal description of the general knowledge and practical methods about visualisation is still lacking in the industry. This renders the clarity of the representation unguaranteed

SemIIM'22: 1st International Workshop on Semantic Industrial Information Modelling, 30th May 2022, Hersonissos, Greece, co-located with 19th Extended Semantic Web Conference (ESWC 2022)

*Corresponding author.
zhuoxun.zheng@de.bosch.com (Z. Zheng); baifanz@ifi.uio.no (B. Zhou)
0000-0002-4223-6746 (Z. Zheng); 0000-0003-3698-0541 (B. Zhou); 0000-0001-6034-4137 (A. Soylu); 0000-0003-3247-4166 (E. Kharlamov)



Procedure:

1. Create **Canvas** with its **figure name** and **layout** (**3 subplots**) to configurate.
2. Represent the first dataset with method **Scatter plot** in subplot 1, with the configuration of **legend** and **axis**.
3. Represent the second dataset with method **Histogram** in subplot 2, with the configuration of **legend** and **axis**.
4. Represent the first dataset with method **Histogram** in subplot 3, with the configuration of **legend** and **axis**.

Figure 1: A visualisation example with its procedure, *1:* Canvas determination; *2,3,4:* subplot drawing

and makes the subsequent discussions on the machine learning results lacks a common basis. In this regard, a visualisation ontology is a good method, which, as formal explicit specifications of shared conceptualisations [11], identifies the general nature and the workflow of visualisation tasks by defining the concepts in the domain and relationships between those concepts. Besides, one can easily extend such visualisation ontology by adding individual information on it and thus have the potential to generate knowledge graphs, which can be able to represent concrete visualisation tasks. Currently there are a few studies that discuss partially the modelling of visualisation. For instance, the computer science ontology [12, 13] contains the general knowledge about visualisation, but the concepts of specific visualisation process is not involved. Statistics ontology [14, 15] enumerates the various visualisation methods, but they insufficiently study the procedures of the visualisation approaches. In conclusion, the existing relevant ontologies are less adequate in depicting the practical procedure of visualisation tasks, which is highly demanded in the Bosch.

To this end, we develop a visualisation ontology and we present our on-going work on this topic. Our visualisation ontology is continuously evaluated and evolved through the common use cases at Bosch, a world leader in automotive industry and Internet of Things. Our studies represent a broad range of visualisation activities. Besides, we align our effort with literature and common programming libraries (e.g. matplotlib in Python). In addition, we also discuss applications and evaluation of our ontology.

2. Our Approach

Visualisation Tasks Rather than being able to represent all the visualisation tasks, *VisuOnto* we present in this short paper aims to cover most of the visualisation practices in data analysis projects at Bosch. Therefore the covered visualisation tasks are limited to charts, that are intended to represent the properties such as the distribution, change, statistical information etc of numerical data. Specific representation methods can be divided into scatter plot, curve plot, histogram, heat map, pie chart, etc. The functions of these chart types can overlap, for example both pie charts and histogram can represent the distribution of a set of data. These types of representation methods we considered can meet the visualisation needs of most data analysis projects at Bosch.

In addition to the distinction in representation methods, the charts can be divided into simple charts and complex charts, A simple chart uses one method to represent one set of data, while a

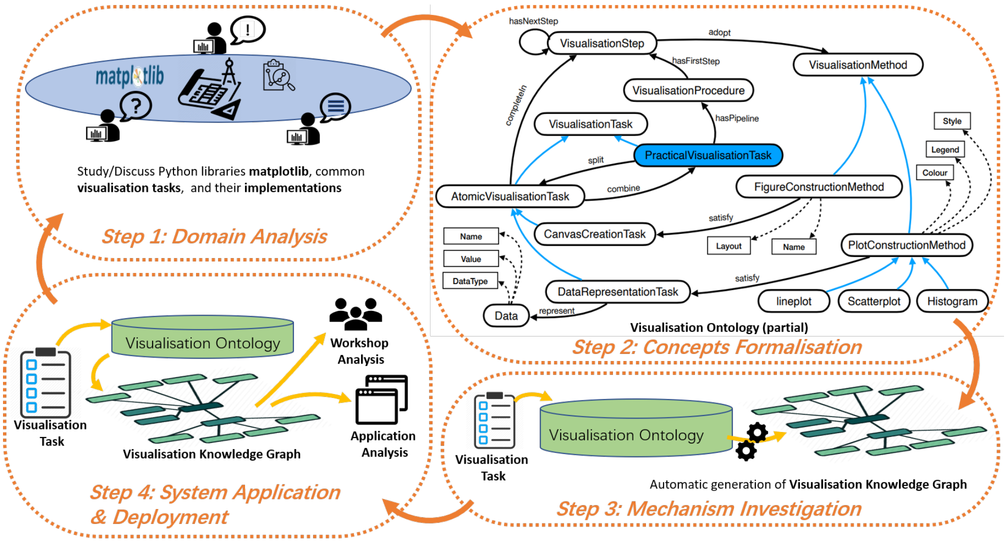


Figure 2: The workflow of generation of visualisation ontology

complex chart uses multiple methods to represent multiple sets of data.

To complete such a visualisation task, i.e., to produce a figure that meets the task requirements, the procedure can be formalised and be divided into such steps: (1) create the canvas of the figure, with the configuration of its name and layout (subplots); (2) in each subplot, represent the desired data according to the customization, which includes the representation method (line plot, scatter plot, etc.) and some details (colour, size, etc.). An example of visualisation with its procedure can be seen in Fig 1.

Three Aspects of Requirements. We now discuss the following aspects of requirements for *VisuOnto*.

R1. Coverage: The ontology should be able to cover the aforementioned visualisation tasks. For the covered types, *VisuOnto* should be able to formalise all the common features. From another point of view, all common features of any diagram of the covered types can be identified by the properties in *VisuOnto*.

R2. Procedure: Ontologies typically contain taxonomies of classes and sub-classes. We also emphasise on the inclusion of *procedures* of visualisation tasks in the modelling. Specifically, our ontology should also reflect the procedure of build a diagram that meets the task requirements. After extending the visualisation ontology into a knowledge graph by adding specific individual information from a concrete visualisation task, one can easily give out the pipeline for that task, which is one of the applications of *VisuOnto* and will in introduced later.

R3. Application: The *VisuOnto* should be as comprehensible as possible, and is thus easy to be used in industry.

Ontology Development Process.

We broadly follow the routine of the Human-Centered Collaborative Ontology Engineering Methodology (HCOME) [16, 17], which is a kind of collaborative ontology engineering methodology. We use Protege as an ontology editor with OWL 2 as the underlying representation language. The whole process can be divided into 4 steps, which are dipicted in Fig 2. *Step 1: Domain Analysis:* We discussed common visualisation tasks at Bosch, read literature, and studied common Python libraries (e.g. matplotlib) in order to comprehend the knowledge of visualisation domain. We enumerate common and important terms of visualisation tasks and classified visualisation tasks into categories to built taxonomies of tasks. In addition, we also studied frameworks of implementing visualisation tasks with popular programming languages (essentially Python). *Step 2: Concepts Formalisation:* Based on the terms collected from the last steps, basic concepts are formalised as classes and relationships between them. *Step 3: Mechanism Investigation.* We study the mechanism of how *VisuOnto* can serve as the basis for our visualisation knowledge graph generation. Visualisation knowledge graph is the knowledge graph representing a concrete visualisation task and also the procedure to solve it by drawing the specific diagram. We will discuss this more in Section 3. *Step 4: System Deployment.* After the validation, the ontology will be deployed in manufacturing, where user feedbacks are collected constantly and lead to further domain analysis and iterative processing.

Visualisation Ontology. The visualisation ontology represents the concept of *building a concrete diagram to present specific data.* Intuitively, to build a diagram with some data to present, one need first to determine the overall properties of the canvas, such as its name and layout. Next, each set of data are presented in the diagram with desired properties. An example of such process in building a diagram can be seen in Fig. 1. According to the requirement of practicability, this ontology , as partially depicted in the right of Fig. 2, emphasis on the workflow of such building process.

Specifically, under the concept of visualisation, there are three classes, *VisualisationTask*, *VisualisationMethod* and *VisualisationProcedure* with their names representing their nature. *VisualisationTask* can be divided into two sub-classes, the *AtomicVisualisationTask* and *PracticalVisualisationTask*. The *AtomicVisualisationTask* models the most basic visualisation components and is thus named as "atomic". There are two kinds (sub-classes) of atomic visualisation tasks, first is *CanvasCreationTask*, which determines the canvas, and the second is *DataRepresentationTask*, which refers the task of presenting one set of data in the canvas accordingly. These two classes of tasks correspond two sub-classes of *VisualisationMethod*, namely *FigureConstructionMethod* and *PlotConstructionMethod* respectively. Another visualisation task, *PracticalVisualisationTask* models the practical visualisation tasks, they can be regarded as the serialization of atomic visualisation tasks. It connects *VisualisationProcedure* with the object property *hasPipeline*. And the class *VisualisationProcedure* consists of a series of *VisualisationStep*, which *adpot* the *VisualisationMethod* and is *completeIn*ed by *AtomicVisualisationTask*. The reasoning of this visualisation ontology includes such constraints: every *PracticalVisualisationTask split* into exactly one *CanvasCreationTask* and at least one *DataRepresentationTask*.

3. Evaluation and Application

In this section we will introduce the evaluations of *VisuOnto* and one of its application.

Workshop Evaluation. To evaluate the developed ontology, several workshops are held in Bosch, In the workshops, practical visualisation tasks in Bosch are collected. Several data scientists and knowledge engineers at Bosch are asked to represent these tasks based on our ontology afterwards. According to the generated representations, data scientists will try to give out the procedural (or python scripts) to complete the tasks. In this process, the questions in three dimensions are studied.

D1: How well can *VisuOnto* represent the collected visualisation tasks.

D2: How well can *VisuOnto* represent the procedure that can be used to complete the collected visualisation tasks.

D3: The hardness to understand and use *VisuOnto*.

These three dimensions of questions correspond to the three aforementioned requirement of our ontology respectively.

Competence Questions. The collected visualisation tasks in the previous step are selected randomly, and knowledge engineers at Bosch encode them into ontologies in instance level (as known as knowledge graphs). Then the competency questions are discussed. The designed competency questions reflect the coverage of the domain knowledge from two aspects in the visualisation, i.e., visualisation tasks inspection (e.g., What are the data desired to represent in one visual-task?), visualisation procedure summary (e.g., What is the last step in drawing a chart for one task?).

Automatic Knowledge Graph Generation. Through our ontology, an knowledge graph representing concrete visualisation tasks can be generated automatically. Specifically, a GUI can be used to ask users give out specific information to describe a visualisation task, those information in individual level can be encoded as the assertional knowledge into the ontology, forming knowledge graphs automatically. Since *VisuOnto* is procedure-orientated, such knowledge graphs not only represent specific visualization tasks, but can also be used to decode specific pipelines to solve the corresponding tasks.

4. Conclusion and Outlook

In this paper we present our ongoing work of visualisation ontology. The generated ontology is easy to understand and covers most of visualisation cases in industry. Additionally, it's practice-orientated, which means this ontology also emphasis on the general knowledge of visualisation pipelines. This ontology is still under evolution in Bosch: it will be continuously evaluated, exploited and utilized in use cases throughout its life cycle, which is part of the future work.

Acknowledgements. The work was partially supported by the H2020 projects Dome 4.0 (Grant Agreement No. 953163), OntoCommons (Grant Agreement No. 958371), and DataCloud (Grant Agreement No. 101016835) and the SIRIUS Centre, Norwegian Research Council project number 237898. We gratefully acknowledge the economic support from The Research Council of Norway and Equinor ASA through Research Council project "308817 - Digital wells for optimal production and drainage" (DigiWell).

References

[1] V. Dhar, Data science and prediction, Communications of the ACM 56 (2013) 64–73.

[2] Z. Zheng, B. Zhou, D. Zhou, G. Cheng, E. Jiménez-Ruiz, A. Soylu, E. Kharlamov, Query-based industrial analytics over knowledge graphs with ontology reshaping, ESWC (Posters & Demos), Springer (2022).

[3] B. Zhou, Z. Zheng, D. Zhou, E. Jimenez-Ruiz, G. Cheng, T. Tran, D. Stepanova, M. H. Gad-Elrab, N. Nikolov, A. Soylu, et al., The data value quest: A holistic semantic approach at bosch, ESWC (Demos/Industry), Springer (2022).
[4] Y. Svetashova, B. Zhou, T. Pychynski, S. Schmidt, Y. Sure-Vetter, R. Mikut, E. Kharlamov, Ontology-enhanced machine learning: a bosch use case of welding quality monitoring, in: International Semantic Web Conference, Springer, 2020, pp. 531–550.
[5] D. Zhou, B. Zhou, J. Chen, G. Cheng, E. Kostylev, E. Kharlamov, Towards ontology reshaping for kg generation with user-in-the-loop: Applied to bosch welding, in: The 10th International Joint Conference on Knowledge Graphs, 2021, pp. 145–150.
[6] D. Zhou, B. Zhou, Z. Zheng, E. V. Kostylev, G. Cheng, E. Jimenez-Ruiz, A. Soylu, E. Kharlamov, Enhancing knowledge graph generation with ontology reshaping–bosch case, ESWC (Demos/Industry), Springer (2022).
[7] M. Yahya, B. Zhou, Z. Zheng, D. Zhou, J. G. Breslin, M. I. Ali, E. Kharlamov, Towards generalized welding ontology in line with iso and knowledge graph construction, ESWC (Posters & Demos), Springer (2022).
[8] C. Naab, Z. Zheng, 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. URL: https://www.sciencedirect.com/science/article/pii/S0921889021001895.
[9] O. Celik, D. Zhou, G. Li, P. Becker, G. Neumann, Specializing versatile skill libraries using local mixture of experts, in: Conference on Robot Learning, PMLR, 2022, pp. 1423–1433.
[10] B. Zhou, D. Zhou, J. Chen, Y. Svetashova, G. Cheng, E. Kharlamov, Scaling usability of ml analytics with knowledge graphs: Exemplified with a bosch welding case, in: The 10th International Joint Conference on Knowledge Graphs, 2021, pp. 54–63.
[11] N. Guarino, D. Oberle, S. Staab, What is an ontology?, in: Handbook on ontologies, Springer, 2009, pp. 1–17.
[12] A. A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, E. Motta, The computer science ontology: a large-scale taxonomy of research areas, in: International Semantic Web Conference, Springer, 2018, pp. 187–205.
[13] K. K. Breitman, M. A. Casanova, W. Truszkowski, Ontology in computer science, Semantic Web: Concepts, Technologies and Applications (2007) 17–34.
[14] K. Kotis, A. Papasalouros, Statistics ontology, 2018. URL: http://stato-ontology.org/.
[15] P. Rocca-Serra, S.-A. Sansone, Experiment design driven fairification of omics data matrices, an exemplar, Scientific Data 6 (2019) 1–4.
[16] K. Kotis, G. A. Vouros, Human-centered ontology engineering: The hcome methodology, Knowledge and Information Systems 10 (2006) 109–131.
[17] G.-B. Alejandra, R.-S. Philippe, Learning useful kick-off ontologies from query logs: Hcome revised, in: 2010 International Conference on Complex, Intelligent and Software Intensive Systems, IEEE, 2010, pp. 345–351.