Knowledge Graph-Based Semantic System for Visual Analytics in Automatic Manufacturing

Baifan Zhou[1], Zhuoxun Zheng[2,3], Dongzhuoran Zhou[2,1], Zhipeng Tan[2,5], Ognjen Savković[4], Hui Yang[6], Yujia Zhang[7] and Evgeny Kharlamov[2,1]

[1] *SIRIUS Centre, Department of Informatics, University of Oslo, Norway*
[2] *Bosch Center for Artificial Intelligence, Germany*
[3] *Department of Computer Science, Oslo Metropolitan University, Norway*
[4] *Department of Computer Science, Free University of Bozen-Bolzano, Italy*
[5] *RWTH Aachen University, Germany*
[6] *Interdisciplinary Laboratory of Digital Sciences, France*
[7] *Department of Electrical and Computer Engineering , University of Alberta, Canada*

Abstract

Visual analytics has been important for many data-driven applications in modern industries.However, there has been limited research of semantic technologies for visual analytics, which hamstrings its transparency and reusability. To this end, we propose a semantic system that encodes visual analytical solutions in reusable knowledge graphs, which can be translated to executable scripts. Further more, our approach incorporates domain knowledge such as feature type information by linking domain ontologies and visual ontology. This poster paper presents preliminary evaluation of our approach with a Bosch use case with promising results.

Keywords

Visual Analytics, Knowledge Graph, KG Generation, Data Science, Manufacturing, Software System

1. Introduction

Visual analytics has been essential for data analysis in a wide range of applications in modern industries, such as in exploratory data analysis for gaining understanding and first insights for the data [1, 2, 3], presenting to the operators for identification of interesting data snippets [4], and representing the results of machine learning analytics [5, 6]. However, there has been limited research of semantic technology on visual analytics, especially in industry [7]. This hamstrings the transparency and reusability, especially for non-data-scientists users, such as engineers, managers, who are important stakeholders in multi-disciplinary industrial projects [8].

Moreover, existing ontologies [9] and tools discussed the general purpose data visualisation, but limitedly exploited domain knowledge (such as feature type information of the data) or the essential procedures for visual analytical scripts [10]. To this end, we propose a semantic system that relies on semantic technologies for user-friendly visual analytics, offering domain knowledge supported visual analytics with no-coding experience. Our approach encodes visual analytical solutions in reusable knowledge graphs (referred to as visual KG) via graphic user interface (GUI) and reasoning, which can be translated to executable scripts [11]. Further more, our approach incorporates domain knowledge in a type of automatic manufacturing,

Hangzhou'22: The 21st International Semantic Web Conference, October 23–27, 2022, Hangzhou, China
baifanz@ifi.uio.no (B. Zhou)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3254/paper359.pdf

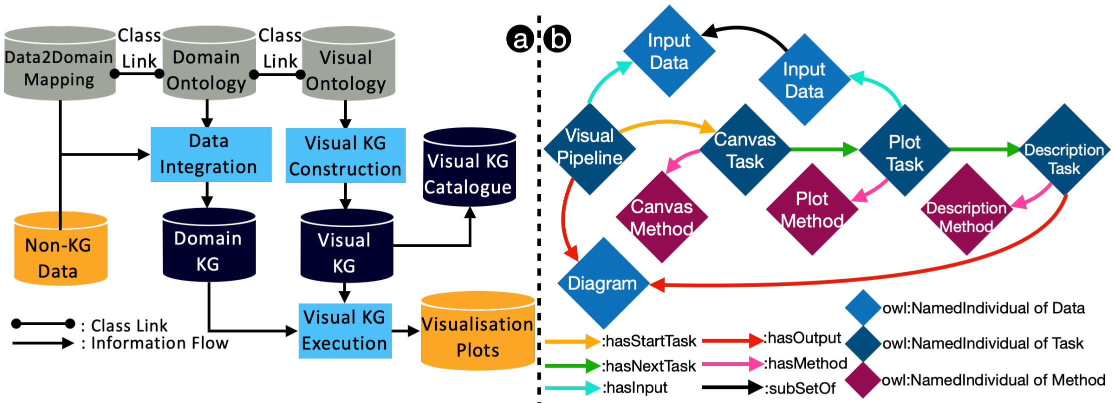

Figure 1: (a) System for semantic visual analytics; (b) Template for Visual KG construction

automated welding by linking domain ontologies and visual ontology (a set of axioms that encode domain knowledge of visual analytics, such as visualisation methods, procedure, and constraints) [12, 13]. In automated welding, robots press metal car body pieces together and a current flows through the robot electrode and the car bodies to melt the metal materials, which generates a welding nugget that connect the metal pieces [14]. This poster paper presents preliminary evaluation of our approach with a Bosch use case which shows promising results. This paper extends the visual KG part of our ISWC full paper about executable KGs [15] by giving more examples and technical details of KG construction and evaluation.

2. Our Approach

System Overview. We now walk through our system (Figure 1a). From the bottom left, the Non-KG Data can be e.g., relational tables (csv), json files. For simplicity, we focus on relational tables. These data are mapped to the domain ontology O^{do} via Data-to-Domain (Data2Domain) Mapping. The Data2Domain Mapping maps tables and table columns to classes in the O^{do}. Through Data Integration, domain KGs are generated. The Visual Ontology O^{visu} contains axioms that describe the knowledge for visual analytics. The Visual KG Construction follows the schema defined by O^{visu} and generates Visual KG [16, 17], either manually [18] by a GUI or automatically following several Visual KG templates (also stored in O^{visu}). The generated Visual KG can be stored in Visual KG Catalogue for later reuse. The Visual KG Execution translates Visual KG to executable scripts and run the scripts, resulting the Visualisation Plots [19, 20].

The Data2Domain Mapping is provided by domain experts in the form of rules as $q_{\text{sq1}}(R) \to A$, where $q_{\text{sq1}}(R)$ is an SQL query on table R that returns table columns and A is a class in the ontologies. We choose OWL 2 QL for O^{do} becasue it is optimised for efficient query answering over relational databases. We choose OWL 2 EL for O^{visu} because of its expressivity and it is still polynomial for query answering. Between Data2Domain Mapping and O^{do} there exist class links, since Data2Domain Mapping maps the Non-KG Data to classes in O^{do}. All classes in O^{do} that represent features in data are linked to classes in O^{visu} via the subclass axioms.

Example 1. *CurrentCurve* is a column in the relational tables *Current* that stores the data of current sensors measured every millisecond. This column is mapped to the class *OperationCurveCurrent* in the O^{do}: SELECT *CurrentCurve* AS *cc* FROM *Current* $\to$ *do:OperationCurveCurrent*(*cc*), which describes the current curve corresponding to a welding operation. The class *OperationCurveCurrent* is linked to the class *TimeSeries* via the subclass axiom: *do:OperationCurveCurrent* $\sqsubseteq$ *visu:TimeSeries*.

Data Integration. With the Data2Domain Mapping and the O^{do}, data from different sources

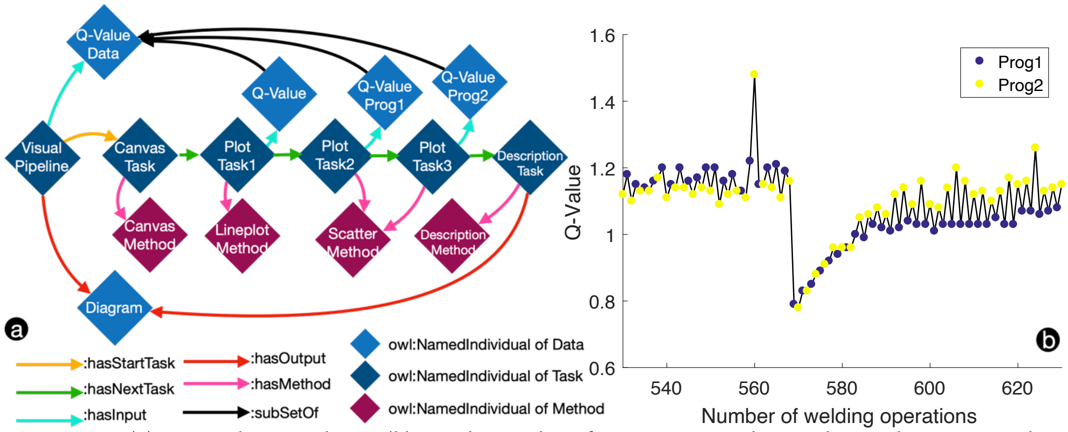

Figure 2: (a) Example Visual KG; (b) resulting plots for inspecting the quality indicator Q-Value.

and formats are integrated into KGs with unified formats and feature names. This is done via the ETL (extract-transform-load) process.

Example 2. A series of columns in different data sources, such as *CurrentCurve*, *Strom*, *CurrentAmp* are all mapped to the class *OperationCurveCurrent* in the O^{do}. They all are renamed to *OperationCurveCurrent* in the resulting Domain KG.

Visual KG Construction. To construct Visual KGs, our approach takes a KG template (Figure 1b), and extends the template with more entities of the *PlotTask* and fills in the properties. We offer two ways of Visual KG construction. One is the manual way done by the users via a GUI (Example 3), or semi-automated via a set of rules with open world assumption (Example 4).

Example 3. Take Fig. 2 as an example. Once the users choose to create a visual KG, the GUI will use the template in Figure 1b, which contains several *owl:NamedIndividuals* of the types *VisualPipeline*, *CanvasTask*, *PlotTask*, and *DescriptionTask*. Next, the users will need to select the input data, and add several entities of *PlotTasks* from available tasks based on O^{visu}. For each *PlotTask*, the input data, the method and some parameters are mandatory to be given. The users need to configure the *PlotTasks* by specifying e.g., the inputs, line colour, line width. After that, they can also configure the *CanvasTask* and the *DescriptionTask* by giving the x label, y label, legend, etc.

Example 4. Our approach first identifies there exists three input features with the labels: *target*, *estimated training* and *estimated test*. Then it generates a KG from the template in Figure 1b with three entities of *PlotTasks* between the *CanvasTask* and *DescriptionTask*. All of them are subclasses of *TimeSeries*. For *TimeSeries*, our approach uses the rule that adopts *LinePlotMethod* as the recommended plot method and adds its typing information: $\exists hasMethod^{-}.(Task \sqcap \exists hasInput.TimeSeries) \sqsubseteq LinePlotMethod$

Visual KG Execution. The execution of visual KGs is language-dependent. Here we use Python for discussion: each individual of *PlotMethod* is a Python script for plotting, whose mandatory inputs/outputs and parameters are clearly defined. Each Visual KG contains an entity of *VisualPipeline* and a series of *Tasks* connected with *hasNextTask*. Thus, the execution of a Visual KG invokes the Python function scripts with the inputs/outputs and parameters given by *Data* and datatype properties of KGs, according to the order defined by *hasNextTask*.

3. Evaluation and Conclusion

Industrial Use Case. We tested our approach in an industrial scenario of machine learning based quality monitoring for automated resistance spot welding at Bosch, which is an impactful

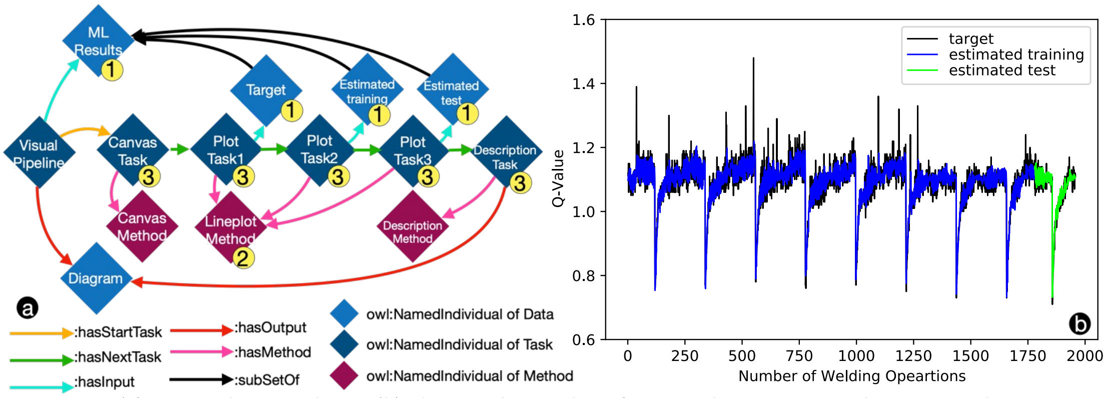

Figure 3: (a) Example Visual KG; (b) the resulting plots for visualising ML prediction results.

welding process accounting for the production of over 50 million cars globally every year. We invited industrial users from Bosch, including non-data-scientists to workshops for evaluation. The visual KGs ran on a sample welding production dataset collected from a factory in Germany, contains 4585 welding operation records.

Visual Analytical Tasks and Reusability. The tasks in the use case can be categorised into two types: I. *Data inspection* (Figure 2a and Example 3), and II. *Results visualisation.* Type I tasks are to inspect various data in exploratory data analysis for understanding the data, and Type II tasks are to visualise results of statistical/ML analytics for discussing and interpreting the statistical/ML models for decision-making.

Example 5. For a new task of Type II and a new dataset in Figure 3a, the users can reuse the *VisualPipeline* in Figure 2a by simply modifying the input data entities (1), the plot methods (2), and properties of the plotting tasks (3, e.g., colour, marker size, labels, legend). This demonstrates the good reusability of our approach.

Table 1: Tasks categories and coverage

Category	Plot types	# KGs	Coverage
Data inspection	Line plot, scatter plot, bar chart	87	100%
	Pie chart	56	85%
	Heatmap	34	85%
Results visualisation	Line plot, scatter plot, bar chart	48	100%
	Heatmap	17	85%

Transparency and Coverage. We programmatically generated 242 KGs and use this as the Visual KG Catalogue. We organised extensive workshops and collected 24 reports from Bosch welding experts, engineers, semantic experts, data scientists. They answered questions such as "*I found the semantic system helps to improve the transparency of visual analytics compared to the case without the approach*", and gave scores ranging 1-5 (from disagree, fairly agree, neural, fairly agree, to agree) which aggregated to 4.28 ± 0.47 (mean ± standard deviation) for the *transparency.* After discussion, we categorised most tasks of visual encountered in our project in groups (Table 1), and give the coverage percentage according to our empirical cases. We can see that most of categories can be covered (above 80%)

Conclusion and Outlook. This poster presents our system of domain knowledge supported KG construction for visual analytics, which offers good reusablity, transparency and coverage for the visual analytic tasks. We evaluated the system on a Bosch welding use case with promising results. As future work we plan to study hierarchical topic modelling to better organise our visual KG catalogue and push the deployment further. We also plan to further improve the system and organise demonstrations to a broader audience [21].

Acknowledgements. The work was partially supported by the H2020 projects Dome 4.0 (Grant

Agreement No. 953163), OntoCommons (No. 958371), and DataCloud (No. 101016835) and the SIRIUS Centre, Norwegian Research Council project number 237898.

References

[1] D. Keim, et al., Visual analytics: Definition, process, and challenges, in: Information visualization, Springer, 2008, pp. 154–175.

[2] C. Naab, et al., Application of the unscented Kalman filter in position estimation a case study on a robot for precise positioning, RobAutonSyst 147 (2022) 103904.

[3] O. Celik, D. Zhou, et al., Specializing versatile skill libraries using local mixture of experts, in: CRL, PMLR, 2022, pp. 1423–1433.

[4] Visual analytics: Scope and challenges, in: Visual data mining, Springer, 2008, pp. 76–90.

[5] A. Chatzimparmpas, et al., A survey of surveys on the use of visualization for interpreting machine learning models, Information Visualization 19 (2020) 207–233.

[6] Z. Zheng, et al., Towards a statistic ontology for data analysis in smart manufacturing, in: ISWC, 2022.

[7] B. Zhou, Z. Zheng, D. Zhou, et al., The data value quest: A holistic semantic approach at bosch, ESWC, Springer (2022).

[8] M. Yahya, et al., Towards generalized welding ontology in line with iso and knowledge graph construction, ESWC, Springer (2022).

[9] A. A. Salatino, et al., The computer science ontology: a large-scale taxonomy of research areas, in: ISWC, Springer, 2018, pp. 187–205.

[10] D. Zhou, et al., ScheRe: Schema reshaping for enhancing knowledge graph construction, in: CIKM, 2022.

[11] D. Zhou, B. Zhou, et al., Towards executable knowledge graph translation, in: ISWC, 2022.

[12] Z. Zheng, B. Zhou, et al., Query-based industrial analytics over knowledge graphs with ontology reshaping, ESWC, Springer (2022).

[13] Z. Zheng, et al., Towards a visualisation ontology for data analysis in industrial applications, in: SemIIM@ESWC, 2022.

[14] B. Zhou, Machine Learning Methods for Product Quality Monitoring in Electric Resistance Welding, Ph.D. thesis, Karlsruhe Institute of Technology, Germany, 2021.

[15] Z. Zheng, et al., Executable knowledge graph for machine learning: A Bosch case for welding monitoring, in: ISWC, 2022.

[16] D. Zhou, B. Zhou, Z. Zheng, et al., Enhancing knowledge graph generation with ontology reshaping–Bosch case, ESWC, Springer (2022).

[17] D. Zhou, B. Zhou, et al., Ontology reshaping for knowledge graph construction: Applied on bosch welding case, in: ISWC, 2022.

[18] D. Zhou, B. Zhou, et al., Towards ontology reshaping for KG generation with user-in-the-loop: Applied to bosch welding, in: IJCKG, 2021.

[19] B. Zhou, et al., Knowledge graph-based semantic system for visual analytics in automatic manufacturing, in: ISWC, 2022.

[20] B. Zhou, Z. Tan, et al., Towards a visualisation ontology for reusable visual analytics, in: IJCKG, 2022.

[21] Z. Zheng, et al., ExeKG: Executable knowledge graph system for user-friendly data analytics, in: CIKM, 2022.