Towards Models of Conceptual and Procedural Operator Knowledge

Richard Nordsieck[1,*], Michael Heider[2], Anton Hummel[1], Alwin Hoffmann[1] and Jörg Hähner[2]

[1]XITASO GmbH IT & Software Solutions, Austraße 35, 86153 Augsburg, Germany
[2]Organic Computing Group, University of Augsburg, Am Technologiezentrum 8, 86159 Augsburg, Germany

Abstract
To increase the utility of semantic industrial information models we propose a methodology to incorporate extracted operator knowledge, which we assume to be present in the form of rules, in knowledge graphs. To this end, we present multiple modelling patterns that can be combined depending on the required complexity. Aiming to combine information models with learning systems we contemplate desired behaviours of embeddings from a predictive quality perspective and provide a suited embedding methodology. This methodology is evaluated on a real world dataset of a fused deposition modelling process.

Keywords
expert knowledge, information model, graph embedding

1. Introduction

On the one hand, standardised semantic information models (IMs) and standards for their hosting, such as the Industry 4.0 asset administration shell, are gaining traction in the industrial internet of things where they can be used to facilitate interoperability and data interchange between different companies, production plants, lines or machines [1, 2, 3]. On the other, knowledge graphs (KGs) are a popular data structure to integrate knowledge of multiple heterogeneous sources [4, 5, 6]. Combined with approaches that allow reasoning over knowledge graphs, e. g. for link prediction to facilitate knowledge graph completion, they are a logical choice for semantic industrial information models [7, 8, 9]. However, the knowledge typically represented in these industrial information models is of mostly factual and conceptual nature, reaching the level of "knowledge of principles and generalizations" of Krathwohl's taxonomy [10]. It concerns equipment [7, 9], material [7], process segments [7], parts [5], products [5, 7], events [9], underlying measurements [11] and geospatial information [12] as well as relations interconnecting these entities. Based on this information, use cases addressed range from

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.
richard.nordsieck@xitaso.com (R. Nordsieck); michael.heider@informatik.uni-augsburg.de (M. Heider); anton.hummel@xitaso.com (A. Hummel); alwin.hoffmann@xitaso.com (A. Hoffmann); joerg.haehner@informatik.uni-augsburg.de (J. Hähner)
0000-0003-3140-1993 (M. Heider); 0000-0003-0107-264X (J. Hähner)

anomaly detection [8] over process monitoring [13, 8] to locating parts and equipment in plants [12] and risk management [13]. We are not aware of the use of industrial IMs containing extracted expert knowledge, which belongs to the more advanced conceptual, i. e. "knowledge of models, theories, structure" [10], and procedural categories of knowledge. This knowledge is of heuristics-like nature and usually obtained through multiple years of expertise. A more detailed explanation of expert knowledge and procedural knowledge for the manufacturing scenario investigated in this paper is given in Section 3.

Since expert knoweldge is playing a crucial role in many industrial day to day processes—from the design of components to reparametrisation processes necessary to deal with quality defects during production—and only available to a limited number of people, representing it in a standardised way would be of great interest for the industry. Therefore, we propose that including procedural knowledge would enhance the applicability of IMs in several ways:

1. a semantic integration between the *what* of given machinery and the *how* to operate it
2. suitability for predictive quality use cases, e. g. by utilizing the resulting knowledge graphs, which would contain quantified knowledge, to increase the performance of learning systems. This could lead to an increase in the ability to generalize and cope with coarse data—both frequent challenges in industrial contexts
3. a standardised representation of procedural knowledge which would (1) enable the creation of digital process twins that could be supplied alongside machinery for operating and training purposes, thereby reducing the impact of changes in a production line, (2) provide a standardised way to combine it with varying kinds of knowledge from different sources, e. g. physical limits of machinery provided by process engineering and (3) enable the fusion of knowledge extracted by traditional [14] as well as data-based [15] methods

Based on an overview of related work (see Section 2) we explore how extracted tacit operator knowledge available as rules can be incorporated in KGs serving as IMs by modelling patterns (Section 3). Section 4 tries to answer the question whether these KGs are able to be embedded in a form that benefits predictive quality use cases. Section 5 provides an outlook describing the next steps towards realizing our vision of knowledge graphs containing extracted expert knowledge as actionable rules while Section 6 concludes this paper.

2. Related Work

Characteristics of established industrial information models and the expected benefits of incorporating rules in these models are described in Section 1. While rules have, to the best of our knowledge, not been directly represented in knowledge graphs used as industrial information models, they have been frequently used in different semantic contexts.

Rule representation in graphs Representations of rules in graphs have been addressed by Chein and Mugnier [16]. They explored how bi-coloured graphs can be used to encode condition and conclusion of rules on both a general level as well as, optionally, for specific entities encoded via attributes. However, their approach does not offer a way to provide quantifications to either conditions or conclusions. Also, established embedding methods are not directly applicable to bi-coloured graphs, which limits the usability of their approach for e. g. knowledge infused learning.

Assisted Embeddings More often than being directly represented in graphs, logical rules are used as auxiliary information for knowledge graph embeddings [17, 18]. Here, rules are either provided by algorithm designers or domain experts to capture common sense knowledge or automatically mined from knowledge graphs [18]. Zhang et al. [19] create relations based on rules that load to an increase in embedding performance. Ringsquandl et al. [20] conclude that the performance of KG completion can be increased by utilizing embeddings of events, i. e. time-series data [9] during KG embedding. However, in contrast to the events or logical rules employed in these approaches, rules founded on extracted operator knowledge frequently contain quantifications of conditions or quantifications which makes them more complex and unsuited to the described approaches.

Rule Representations in Embeddings It has been shown that "existential rules can be exactly represented using convex regions of knowledge graph embeddings" [21]. While a methodology that provides exact representation of general rules in embeddings would provide greatly helpful in evaluating the suitability of different methodological choices of rule representation in knowledge graphs we cannot rely on Gutierrez's methodology since the rules containing the experts' extracted knowledge are more complex than the existential rules considered. Furthermore, representing rules in knowledge graphs as opposed to embeddings, provides a significant benefit for information models as the representation is more direct and can be independently accessed.

Establishing Embedding Quality Link prediction and entity classification are the standard scenarios to evaluate embedding methods [22, 23, 24, 25, 19, 26, 27, 28, 29]. However, doubt has been cast both on biases in the used datasets [30, 31, 32] as well as on the more general capability of KG embeddings to capture semantics [33]. Therefore, behavioural testing of embedding methodologies is gaining attention [34]. Since we are aiming at using embeddings not only for link prediction but to improve the performance of learning systems, an evaluation of the embeddings' encoding of the required semantic information is necessary in our case.

3. Representations of Operator Knowledge in Industrial Information Models

Based on a manufacturing scenario, this section will introduce modelling patterns for different representations of expert knowledge along with an overview of their properties.

3.1. Representations

In manufacturing use cases, a high proportion of expert knowledge is tacit operator knowledge which pertains to parametrisation of machinery. As such, it contains knowledge about both conceptual relationships between process parameters and quality characterisitcs as well as procedural behaviours that lead to the achievement of goals, i. e. how and in what order to adapt process parameters to mitigate occurring quality defects and achieve a perfect parametrisation. In this paper we focus on the aspect of how parameters are adjusted. This tacit operator knowledge can be extracted by various methodologies [e.g. 14, 15] leading to rules at different levels of abstraction. Also, information concerning the same problem might be available from alternative sources such as process engineering documents or handbooks which could be combined or contrasted with knowledge operators gained in practice. As such, adapting and building on definitions for operator knowledge presented in [35], we inspect several modelling patterns for representing the underlying knowledge at different abstraction levels that can be combined at will. In the following, we will align our terminology with manufacturing scenarios to increase the readability of examples.

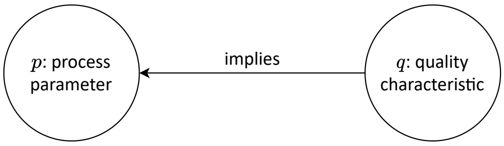
From these modelling patterns, a fitting degree of abstraction can be chosen to either reflect the kind of knowledge that is available or of the information models' specific domain. This is relevant since we expect that the higher complexity, i. e. through hierarchies, provides challenges for embedding approaches. Avoiding unneeded complexity in the representation is therefore likely to achieve better results. As such, we recommend choosing the highest abstraction level, that is able to encode all present information. Note that the representations of higher abstraction levels can be easily converted to representations of lower abstraction levels. Therefore including operator knowledge of a different source which utilises a different abstraction level is still possible.

3.1.1. Unquantified Rules

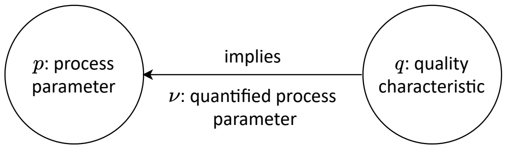
A rule at the highest level of abstraction, i. e. an unquantified rule, could be verbalised as *If quality characteristic q is unsatisfactory then adjust process parameter p*. It can be viewed as an *implies* relation between the condition *quality characteristic q* and the conclusion *parameter p*. This corresponds to the triple notation of (head, relation, tail) common in knowledge graphs. Adapting the definitions of [35] we define parameters $p \in P$ and quality characteristics $q \in Q$. This yields the triple $r_\eta = (q, \langle\text{implies}\rangle, p)$. Whereas in [35] an index for the process iteration was included, we omit it here for the sake of readability as it is not relevant for the contents of this paper. A graphical representation of r_η is shown in Figure 1a. A modelling alternative would be a *is a* relation between q and the semantic meaning of quality characteristics. However, this semantic information is already encoded by the directed relation. Making it explicit would only increase syntactic complexity and hierarchy.

3.1.2. Quantified Conclusions

Rules with quantified conclusions, i. e. parameters, $r_{\hat{\rho}}$ can be verbalised as *If quality characteristic q is unsatisfactory then adjust process parameter p by* ν with $\nu \in \mathbb{R}$. Therefore, $r_{\hat{\rho}} = (\eta_{q,p}, \nu) = (q, \langle\text{implies}\rangle, p, \nu)$. Generally, we want to keep the representation as succinct as possible. Therefore, the representation of unquantified rules is extended to use ν as a weight of the *implies* relation (cf. Figure 1b). Here, the quantified parameter $\rho \in \mathrm{P}$, where $\mathrm{P} = \{(\nu, \langle\text{quantifies}\rangle, p) \mid p \in P \text{ and } \nu \in \mathbb{R}\}$ is implicitly modelled. We denote the rules of quantified conclusions as $r_{\hat{\rho}}$ since several observations are aggregated into the quantification.



(a) Graphical representation of r_η, i. e. *implies* relation between a quality characteristic and parameter.



(b) Graphical representation of $r_{\hat{\rho}}$ with weighted *implies* relation.

Figure 1: Graphical representation of modelling patterns for unquantified rules r_η and rules with quantified conclusions $r_{\hat{\rho}}$.

3.1.3. Quantified Conditions

In addition to quantified parameters that serve as actionable recommendations for operators, the conditions, i. e. quality characteristics, can also be quantified to arrive at more descriptive rules. With quantified conditions, it is possible to represent more advanced concepts, e. g. for higher defects in a specific quality characteristic, parameters need to be adjusted more substantially. While in theory quantified quality characteristics could be used without quantified parameters, it is not beneficial in practice since the conclusion of the rule would remain the same. As such, we consider quantified parameters as a prerequisite of quantified quality characteristics. Rules with aggregated quantified quality characteristics and parameters $r_{\hat{o},\hat{\rho}}$ can therefore be verbalised as *If quality characteristic q is within μ, then adjust process parameter p by ν*, where $\mu \in [g, h]$. This results in the 5-tuple $r_{\hat{o},\hat{\rho}} = (q, \mu, \langle\text{implies}\rangle, p, \nu)$, that can be represented as shown in Figure 2. Here, an explicit modelling of quantified parameters $\mathrm{o} \in \mathrm{O}$, with $\mathrm{O} = \{(\mu, \langle\text{quantifies}\rangle, q) \mid q \in Q \text{ and } \mu \in \mathbb{R}\}$ becomes necessary.

Strictly speaking this leads to a further indirection, since the *implies* relation now connects the actual value of quantified parameter and quality characteristic, which are decoupled from their semantic interpretation by a *quantifies* relation. Transformed into hierarchical triples this yields:

$$r_{\hat{o},\hat{\rho}} = ((\mu, \langle\text{quantifies}\rangle, q), \langle\text{implies}\rangle, (\nu, \langle\text{quantifies}\rangle, p))$$

3.1.4. Multiple Conditions

Sometimes a specific parametrisation is only relevant if multiple conditions align. These rules, r_{q^n}, could be verbalised as *If quality characteristic x and quality characteristic z are unsatisfactory, then adjust process parameter p.* For this modelling pattern, we omitted quantifications for the sake of brevity. r_{q^n} could be trivially encoded by having multiple separate rules for each quality characteristic influencing the same process parameter in the IM, e. g. $r_{\eta_{q,p}}$ and $r_{\eta_{s,p}}$, with $q, s \in Q$ and $p \in P$. However, in this case it would not be clear whether the rules are related according to a logical *AND*, *OR*, or a different operator altogether. As such, we propose the introduction of a relator vertex representing the logical operator required by the r_{q^n} in question. For the example of an *AND-relator* shown in Figure 3 this yields $r_{q^n} = \{(q, \langle\text{implies}\rangle, p), (s, \langle\text{implies}\rangle, p)\}_{\text{AND}}$, where $\{\}_{\text{AND}}$ denotes the set of all relations combined by the respective *AND-relator* l. This can

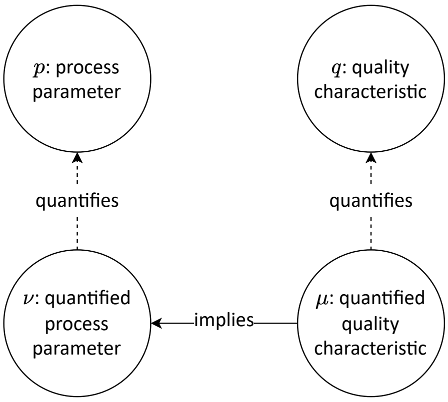


Figure 2: Graphical representation of $r_{\hat{o},\hat{\rho}}$ with explicit modelling of quantified parameter and quality.

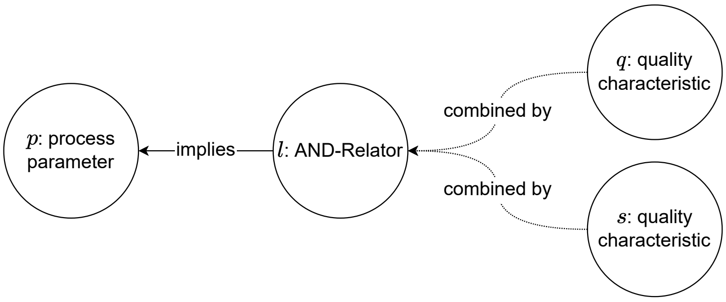


Figure 3: Graphical representation of modelling pattern r_{q^n}, i. e. an *implies* relation indirectly spanning multiple conditions, that are connected by a relator vertex.

be expanded to the following hierarchical triple:

$$r_{q^n} = \{((q, \langle\text{combined by}\rangle, l), \langle\text{implies}\rangle, p), ((s, \langle\text{combined by}\rangle, l), \langle\text{implies}\rangle, p)\}_{\text{AND}}$$

In theory, there is no limit to the number of relations being combined by a relator, the definitions here are based on the example in Figure 3.

The case of one condition influencing several conclusions can be unambiguously represented by adding a separate rule for each of the conclusions. Therefore, this case does not require a special relator vertex.

3.1.5. Inclusion of Process Data

Noy et al. make the point that "it is critical not to lose the linkage between the relationships stored in the graph and where those relationships come from" [5]. While they refer to the discovery process, we assume that capturing semantics of operator knowledge in IMs could be aided by including process knowledge, especially since Ringsquandl et al. [20] have achieved promising results in knowledge graph completion by considering event embeddings. In manufacturing, orders Ω, which can be viewed as compositions of process iterations I, are produced for certain amounts of time. We propose to explicitly model this process data as

$(i, \langle\text{belongs to}\rangle, \omega)$, where $i \in I$ and $o \in O$. The process iterations can be connected with the resulting quantified parameters, $(\rho_i, \langle\text{chosen in}\rangle, i)$, where $\rho \in \mathrm{P}$, quantified quality characteristics $(a_i, \langle\text{is exhibited after}\rangle, i)$, and quantified influences $(b_{i-1}, \langle\text{influences}\rangle, i)$, where $a, b \in \mathrm{O}$ and $(b_{i-1}, \langle\text{is exhibited after}\rangle, i-1)$. P and O are defined in the modelling patterns for *quantified parameters* and *quantified conditions*, respectively. We note that process iteration i is preceded by $i-1$ in order ω, i. e. the quality characteristic b_{i-1} is exhibited after the conclusion of the preceding process iteration.

Connecting the process data with the modelling patterns described above is beneficial, especially in the case of data-based knowledge extraction, since it allows explanation by example. The definition is analogous for parameters p. Using this, we can denote a relationship between process data and aggregate as $(a_i, \langle\text{contributes to}\rangle, \hat{a})$, where $\hat{a}$ is an aggregation of quantified quality characteristic expressed in a specific rule. Both, a_i and $\hat{a}$, are expressed values of a quality characteristic and as such are equivalent in regards to their hierarchical level. The only difference is that the value of a_i is sampled from the real world process whereas $\hat{a}$ is calculated based on a set of observations.

If all process data is included in the IM, a *supports* relation could also be defined following the quality metric used in rule mining [36, 37]. However, since data-based rules rely on aggregates, a direct comparison between $\hat{q}_j$ and q_i, as well as $\hat{p}_j$ and p_i would fail and need to be relaxed by an interval in which they are considered equal. Also, since the rules are split between parameters and quality characteristics there would have to be two separate relations.

3.2. Properties

Here, we will give an overview of the properties shown by different modelling patterns. Firstly, with exceeding expressiveness of and information contained in the representation its complexity is increasing. This increase of abstraction can be seen in the increase in hierarchy hierarchy for each additional quantification or multiple conditions. Secondly, if process data is included in the IM it is likely to lead to a strong imbalance, since process data is much more readily available than extracted operator knowledge. Both aspects highlight the challenges embedding methodologies for IMs with operator knowledge have to address. As most of the relations described in Section 3 are not symmetric it would be easy to generate inverse relations, e. g. *implied by* for *implies*, which could be beneficial in knowledge graph completion settings.

4. Embedding Industrial Information Models containing Operator Knowledge

In this section we aim to give a first indication whether it is possible to embed knowledge graphs containing extracted operator knowledge available as rules. As such, we present a first step towards a methodology for constructing such an embedding and provide a preliminary evaluation. To this end, we utilize the dataset presented in [35] of a fused deposition modelling (FDM) process. We choose to model the knowledge with the pattern of *quantified conclusions*, since the dataset does not provide the data for more complex patterns, i. e. quantified conditions.

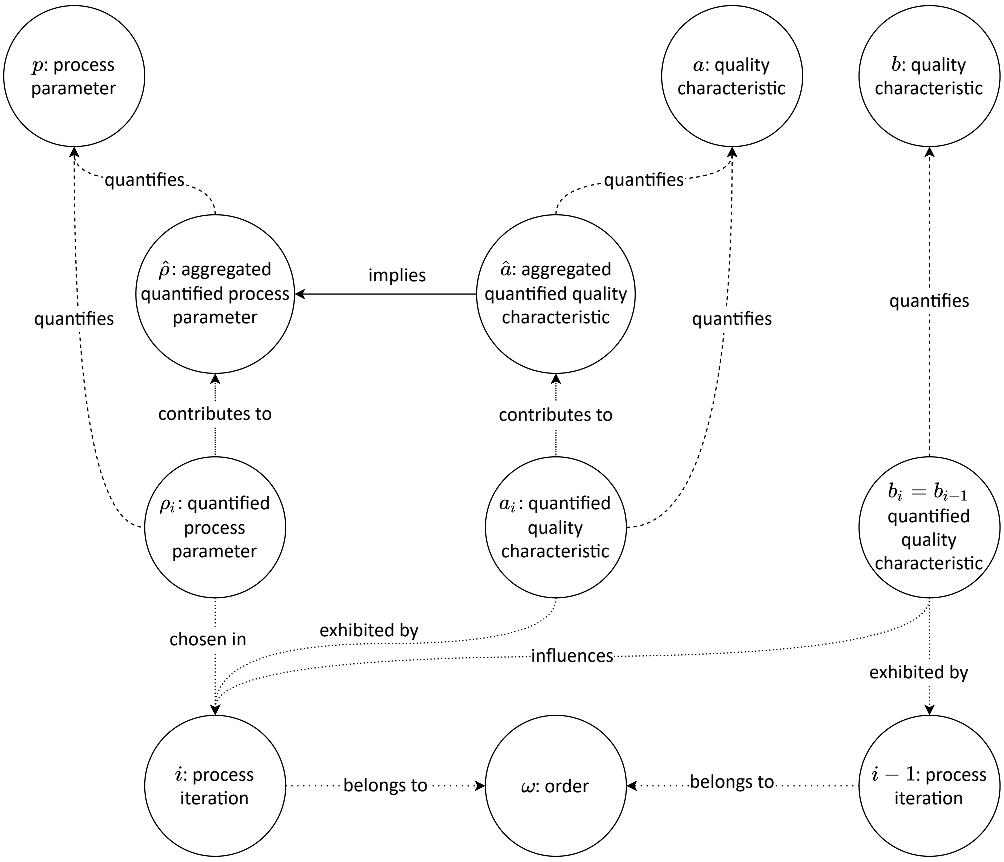


Figure 4: Information Model for process data, i. e. parametrisation processes and their iterations, as well as a rule. The rule's condition and conclusion is quantified relying on aggregations of quantified process parameters and quality characteristics that result from the iteration processes.

The accompanying code is available on github[1].

4.1. Embedding Methodology

To embed operator knowledge that is intended to assist learning systems instead of knowledge graph completion, subgraphs containing the knowledge that is particularly relevant for the given input should be embedded. In our case, the input is a defective quality characteristic, e. g. stringing, a common problem in FDM, that can be alleviated through a fitting parametrisation by the operator or learning system. Our embedding methodology (cf. Figure 5) is loosely based on the methodology outlined in Kursuncu et al. [38] that is an approach towards embeddings for learning systems addressing classification in an natural language processing (NLP) setting, rather than KG completion or predictive quality scenarios.

We follow a sum-based approach [39] that aggregates individual node embeddings, that are particularly relevant to the input quality characteristic. To identify the fitting subgraph $\mathcal{S}$, the

[1] https://github.com/0x14d/embedding-operator-knowledge

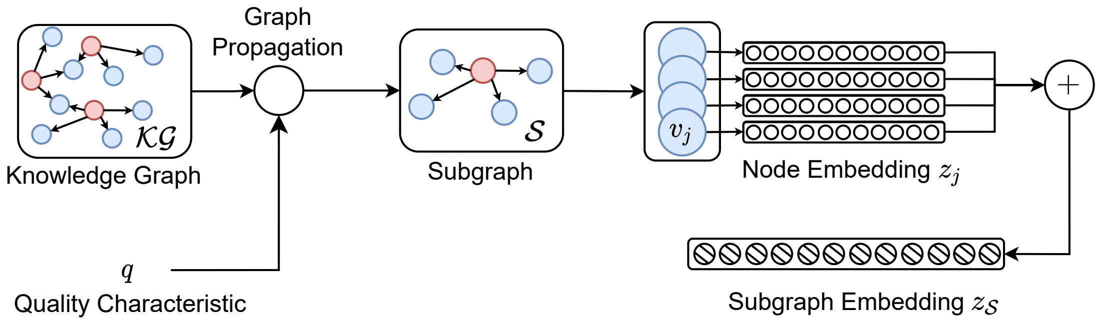


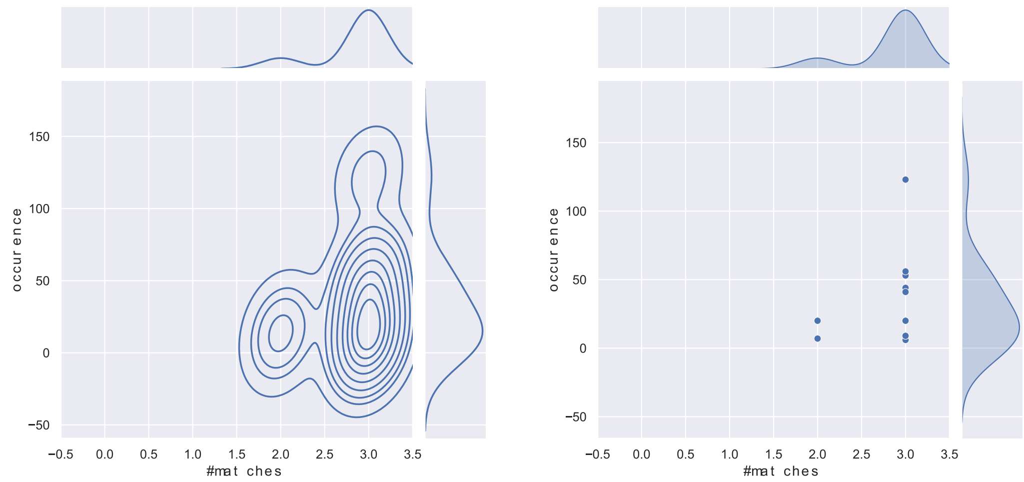
Figure 5: Illustration of the embedding methodology. The quality characteristic q is used to determine the starting node to propagate from. Each parameter node of the resulting subgraph $\mathcal{S}$ then individually embedded as node embedding, e. g. parameter node v_j is embedded as z_j. The parameter node embeddings are then aggregated to one subgraph embedding $z_{\mathcal{S}}$.

input is mapped to the respective node in the knowledge graph. Then, this node is propagated by one step for all outgoing edges to arrive at the parameters adjusted to alleviate this quality defect. Based on this, the propagated nodes, embedded by TransH [29], z are aggregated by $z_{\mathcal{S}} = \sum_{v_i, v_j \in \mathcal{S}} z_j \otimes D(z_i, z_j)$ to form the subgraph embedding $z_{\mathcal{S}}$. Here, v_i, v_j are the pairs of head and tail nodes resulting from the graph propagation and $D(z_i, z_j)$ is the euclidean distance between the node embeddings of v_i and v_j. Dependent on which semantic information should be represented in the subgraph embedding, it must be decided which node embeddings to aggregate in $z_{\mathcal{S}}$. If the head node does not hold semantic information, we suggest ignoring the head node in the subgraph embedding. As this is the case in our scenario, we only aggregated the node embeddings for the parameters. If a modelling pattern for a different abstraction level is used, e. g. *quantified conditions*, the propagation step has to be increased to deal with the introduced indirections.

4.2. Evaluation

4.2.1. Evaluation Metric

Metrics commonly used to evaluate embeddings in knowledge graph completion settings, e. g. *mean reciprocal rank* and *hits@k*, are unsuited to establish the quality of embeddings of operator knowledge in our scenario since the required ground truth is not present. Instead, we propose that the fundamental behaviour of rule embeddings in predictive quality scenarios should be that the parameters adjusted for similar quality characteristics in the (sub)graph, should be as equal as possible to those in the embedding space. Therefore, we define a metric in analogy to *hits@k*, *matches@k*, based on the amount of overlap between the K closest quality characteristics in embedding and graph space. To be able to establish the amount of overlap, a set of the K closest quality characteristics is prepared by ordering them descendingly according to their similarity—amount of overlapping parameters adjusted (higher is better) and euclidean distance (lower is better) for graph and embedding space, respectively. Then, the number of



(a) Two dimensional kernel density estimate plot for #matches and occurrence.

(b) Scatterplot showing the actual results achieved for the respective quality characteristics

Figure 6: All quality characteristics evaluated by *matches@K* for $K = 3$ against their respective occurrences.

matches, #matches, between the respective sets is calculated for each quality characteristic. By conducting the comparison on a set, we ensure that small differences in similarity are not unduly exaggerated in the overall metric since the order is not important to determine a match.

For *matches@k*, K has to be chosen according to the respective dataset. It decreases in expressiveness with increasing size since the unordered nature of the comparison leads to number of matches equalling the number of quality characteristics $|Q|$ if $K = |Q|$, which would equal an overlap of $100\,\%$. Therefore, inspecting the actual similarities between quality characteristics in the graph space is necessary to determine the K which is representative for real world similarity. This can either be done by relying on domain knowledge or by determining the point at which the similarities are abruptly decreasing. In the following experiment we use $K = 3$ since we established experimentally that for greater K the similarities between the quality characteristics are rapidly increasing.

4.2.2. Experimental Evaluation

To determine whether it is possible to embed knowledge graphs containing explicit operator knowledge we conduct an experiment on the dataset described in [35]. After preprocessing and removing categorical parameters, it contains ratings of 13 quality characteristics and a total of 46 parameters that are adjusted to optimise the quality characteristics. The distribution of quality characteristics is skewed with more operator knowledge being present for those that occur more often.

Applying the methodology and metric described above we receive a mean #matches of 2.85 ± 0.38 ($95.00\,\% \pm 12.67\,\%$) for $K = 3$ over all quality characteristics for 46 dimensional embeddings. While this indicates a relatively high overlap, we investigate its distribution for the individual quality characteristics combined with their occurrence in Figure 6. In Figure 6a we can see that that the performance seems to generally increase with increasing occurrence of quality characteristics in the dataset. However, there seems to be a second cluster of well performing quality characteristics with relatively low occurrence that yields good results. Inspecting Figure 6b confirms this notion. Since we assume more operator knowledge to be present in the graph for quality characteristics with higher occurrence the fact that higher occurring quality characteristics lead to better results seems to underpin the conclusion that extracted operator knowledge in embeddings can be represented by embeddings. However, a significant portion of quality characteristics with low occurrence also leads to good results.

5. Future Work

While the presented preliminary evaluation strengthens our hypothesis, a more thorough evaluation is needed to arrive at a firm conclusion. This includes a comparison of the proposed embedding method on the different modelling patterns. Also, the influence of increasing hierarchies, due to increasing complexity, on embedding methods will be investigated. In this context, evaluating more complex embedding methods which have been shown to deal well with hierarchies between concepts such as RotH [23] would be interesting. Moreover, an evaluation on multiple datasets would allow greater confidence in regards to the transferability of the described concepts. However, we are not aware of any suitable public datasets in the industrial domain at this time.

Furthermore, the behaviour of the embedding methods for varying levels of noise in the data should be investigated, since complex information models are rarely error free. Additionally, uncertainties of operators could be encoded using soft rules.

In the patterns modelling operator knowledge presented in this work we strongly relied on relations to represent properties of vertices and relations. These properties could also be represented as attributes of vertices. While this would reduce the involved hierarchies it imposes other complexities for embedding methodologies. As such the integration of attribute-based embeddings [40, 41] could be beneficial.

In addition, the applicability of common KG completion approaches on KGs containing operator knowledge could be researched to infer relations or nodes that have not been encountered in reality, thereby increasing the information content of the representation.

Lastly, by an integration with learning systems, as outlined by Kursuncu et al. [38] for NLP, we could directly measure the impact of the knowledge contained in IMs on the predictive power of learning systems.

6. Conclusion

In this paper we presented several modelling patterns for including extracted operator knowledge into industrial information models, represented as knowledge graphs. These modelling patterns can be conceived as architectural patterns and can be combined and applied depending on the required complexity that should be expressed. Furthermore, we presented an embedding methodology to represent this knowledge as a vector that could be used to combine learning systems with operator knowledge. We established a metric suited to evaluate the embedding's capability to capture semantic relations between conditions, i. e. quality characteristics, based on their resulting conclusions, i. e. parametrisations. In a preliminary evaluation, we have shown that the chosen information model and the proposed embedding methodology are able to express and capture semantic relationships between conditions that lead to similar conclusions if they occurred in a sufficient quantity.

Acknowledgments

This work is funded by the Bavarian Ministry of Economic Affairs, Energy and Technology in the scope of the ADELeS project.

References

[1] M. Platenius-Mohr, S. Malakuti, S. Grüner, T. Goldschmidt, Interoperable digital twins in iiot systems by transformation of information models: A case study with asset administration shell, in: Proceedings of the 9th International Conference on the Internet of Things, 2019, pp. 1–8.

[2] S. Beden, Q. Cao, A. Beckmann, Semantic asset administration shells in industry 4.0: A survey, in: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), 2021, pp. 31–38.

[3] W. P. von Pilchau, V. Gowtham, M. Gruber, M. Riedl, N.-S. Koutrakis, J. Tayyub, J. Hähner, S. Eichstädt, E. Uhlmann, J. Polte, et al., An architectural design for measurement uncertainty evaluation in cyber-physical systems, Annals of Computer Science and Information Systems 22 (2020) 53–57.

[4] G. Buchgeher, D. Gabauer, J. Martinez-Gil, L. Ehrlinger, Knowledge graphs in manufacturing and production: A systematic literature review, IEEE Access 9 (2021) 55537–55554.

[5] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: Lessons and challenges, Communications of the ACM 62 (2019) 36–43. doi:10.1145/3331166.

[6] D. Obraczka, A. Saeedi, E. Rahm, Knowledge graph completion with famer, Proc. DI2KG (2019).

[7] I. Grangel-González, F. Lösch, A. ul Mehdi, Knowledge graphs for efficient integration and access of manufacturing data, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), volume 1, 2020, pp. 93–100.

[8] E. G. Kalaycı, I. Grangel González, F. Lösch, G. Xiao, E. Kharlamov, D. Calvanese, et al., Semantic integration of bosch manufacturing data using virtual knowledge graphs, in: International Semantic Web Conference, 2020, pp. 464–481.

[9] M. Ringsquandl, S. Lamparter, R. Lepratti, P. Kröger, Knowledge fusion of manufacturing operations data using representation learning, in: IFIP International Conference on Advances in Production Management Systems, 2017, pp. 302–310.

[10] D. R. Krathwohl, A revision of bloom's taxonomy: An overview, Theory into practice 41 (2002) 212–218.

[11] M. Ringsquandl, S. Lamparter, R. Lepratti, Graph-based predictions and recommendations in flexible manufacturing systems, in: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, 2016, pp. 6937–6942.
[12] N. Petersen, L. Halilaj, I. Grangel-González, S. Lohmann, C. Lange, S. Auer, Realizing an rdf-based information model for a manufacturing company–a case study, in: International semantic web conference, 2017, pp. 350–366.
[13] T. Hubauer, S. Lamparter, P. Haase, D. M. Herzig, Use cases of the industrial knowledge graph at siemens, in: International Semantic Web Conference (P&D/Industry/BlueSky), 2018.
[14] L. Hörner, M. Schamberger, F. Bodendorf, Externalisierung von prozess-spezifischem mitarbeiterwissen im produktionsumfeld, Zeitschrift für wirtschaftlichen Fabrikbetrieb 115 (2020) 413–417.
[15] R. Nordsieck, M. Heider, A. Winschel, J. Hähner, Knowledge extraction via decentralized knowledge graph aggregation, in: 2021 IEEE 15th International Conference on Semantic Computing (ICSC), 2021, pp. 92–99.
[16] M. Chein, M.-L. Mugnier, Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs, Springer eBook Collection Computer Science, Springer London, London, 2009. doi:10.1007/978-1-84800-286-9.
[17] S. Guo, Q. Wang, L. Wang, B. Wang, L. Guo, Jointly embedding knowledge graphs and logical rules, in: Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 192–202.
[18] S. Guo, Q. Wang, L. Wang, B. Wang, L. Guo, Knowledge graph embedding with iterative guidance from soft rules, Proceedings of the AAAI Conference on Artificial Intelligence 32 (2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11918.
[19] W. Zhang, B. Paudel, L. Wang, J. Chen, H. Zhu, W. Zhang, A. Bernstein, H. Chen, Iteratively learning embeddings and rules for knowledge graph reasoning, in: The World Wide Web Conference, 2019, pp. 2366–2377.
[20] M. Ringsquandl, E. Kharlamov, D. Stepanova, M. Hildebrandt, S. Lamparter, R. Lepratti, I. Horrocks, P. Kroeger, Event-enhanced learning for knowledge graph completion, 2018.
[21] V. Gutiérrez-Basulto, S. Schockaert, From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules, in: Sixteenth International Conference on Principles of Knowledge Representation and Reasoning, 2018.
[22] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in Neural Information Processing Systems 26 (2013).
[23] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, C. Ré, Low-dimensional hyperbolic knowledge graph embeddings, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics 58 (2020) 6901–6914. URL: https://aclanthology.org/2020.acl-main.617.pdf.
[24] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: International conference on machine learning, 2016, pp. 2071–2080.
[25] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, arXiv preprint arXiv:1902.10197 (2019).

[26] H. Yang, L. Zhang, B. Wang, T. Yao, J. Liu, Cycle or minkowski: Which is more appropriate for knowledge graph embedding?, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 2301–2310.
[27] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. den van Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European semantic web conference, 2018, pp. 593–607.
[28] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Twenty-ninth AAAI conference on artificial intelligence, 2015.
[29] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 28, 2014.
[30] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Transactions on Knowledge Discovery from Data (TKDD) 15 (2021) 1–49.
[31] A. Mohamed, S. Parambath, Z. Kaoudi, A. Aboulnaga, Popularity agnostic evaluation of knowledge graph embeddings, in: Conference on Uncertainty in Artificial Intelligence, 2020, pp. 1059–1068.
[32] P. Tabacof, L. Costabello, Probability calibration for knowledge graph embedding models, arXiv preprint arXiv:1912.10000 (2019).
[33] N. Jain, J.-C. Kalo, W.-T. Balke, R. Krestel, Do embeddings actually capture knowledge graph semantics?, in: European Semantic Web Conference, 2021, pp. 143–159.
[34] W. Ben Rim, C. Lawrence, K. Gashteovski, M. Niepert, N. Okazaki, Behavioral testing of knowledge graph embedding models for link prediction, in: 3rd Conference on Automated Knowledge Base Construction, 2021.
[35] R. Nordsieck, M. Heider, A. Hoffmann, J. Hähner, Reliability-based aggregation of heterogeneous knowledge to assist operators in manufacturing, in: 2022 IEEE 16th International Conference on Semantic Computing (ICSC), 2022, pp. 131–138.
[36] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, X. Ren, Jointly learning explainable rules for recommendation with knowledge graph, in: The world wide web conference, 2019, pp. 1210–1221.
[37] V. T. Ho, D. Stepanova, M. H. Gad-Elrab, E. Kharlamov, G. Weikum, Rule learning from knowledge graphs guided by embedding models, in: International Semantic Web Conference, 2018, pp. 72–90.
[38] U. Kursuncu, M. Gaur, A. Sheth, Knowledge infused learning (k-il): Towards deep incorporation of knowledge in deep learning, Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning and Knowledge Engineering in Practice (AAAI-MAKE) (2020).
[39] W. L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: Methods and applications, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 40 (2017) 52–74.
[40] Z. Sun, Q. Zhang, W. Hu, C. Wang, M. Chen, F. Akrami, C. Li, A benchmarking study of embedding-based entity alignment for knowledge graphs, Proceedings of the VLDB Endowment 13 (2020) 2326–2340. doi:10.14778/3407790.3407828.

[41] Z. Sun, W. Hu, C. Li, Cross-lingual entity alignment via joint attribute-preserving embedding, in: International Semantic Web Conference, 2017, pp. 628–644.