Knowledge extraction via decentralized knowledge graph aggregation

Richard Nordsieck, Michael Heider, Anton Winschel, Jörg Hähner

Angaben zur Veröffentlichung / Publication details:

Nordsieck, Richard, Michael Heider, Anton Winschel, and Jörg Hähner. 2021. "Knowledge extraction via decentralized knowledge graph aggregation." In 2021 IEEE 15th International Conference on Semantic Computing (ICSC), 27-29 January 2021, Laguna Hills, CA, USA, edited by Dick Bulterman, Atsushi Kitazawa, David Ostrowski, Phillip Sheu, and Jeffrey Tsai, 92–99. Piscataway, NJ: IEEE. https://doi.org/10.1109/icsc50631.2021.00024.

Knowledge Extraction via Decentralized Knowledge Graph Aggregation

Richard Nordsieck*, Michael Heider†, Anton Winschel* and Jörg Hähner†
**XITASO GmbH IT & Software Solutions*, Augsburg, Germany
{richard.nordsieck, anton.winschel}@xitaso.com
†*Organic Computing Group*, *University of Augsburg*, Augsburg, Germany
{michael.heider, joerg.haehner}@informatik.uni-augsburg.de

***Abstract*—In many industrial manufacturing processes, human operators play a central role when it comes to parameterizing the involved machinery and dealing with errors in the process. However, large parts of the acquired process knowledge are tacit, leading to difficulties sharing the knowledge between operators. Therefore, knowledge extraction is a necessary but time and cost intensive process, requiring both specially trained personnel and experienced operators. In contrast, we propose that by gathering insights into what influenced operators' actual parameter choices, tacit process knowledge can be extracted during production in an example-based manner. This decentralized knowledge—decentralized in regards to who holds knowledge and where it was extracted—is then aggregated to a coherent knowledge graph. We showcase our methodology on a real-world dataset in the domain of fused deposition modeling (FDM), which is generated by operators providing their insights without additional assistance using extended human machine interfaces. Furthermore, we compare rules extracted from the aggregated knowledge graph against an established FDM knowledge base showing the viability of our approach even with limited amounts of data.**

I. Introduction

Autonomous processes for machines or production lines is of increased interest for the manufacturing industry [1]. In many cases, either the details of the manufacturing proess are not fully understood which prevents precise mathematical modeling or explicit model creation is not possible or cost effective. This necessitates addressing the task of (re-) parameterization, i.e. finding a parameterization for which the manufacturing process produces satisfactory results. Parameterization is currently mostly done by experienced operators following an iterative time and resource intensive workflow, which is similar among different manufacturing processes (cf. Figure 1). At first, the manufacturing process, which is influenced by environmental influences, is executed with a standard parameterization. Then, the quality of produced parts is evaluated in regard to specific target criteria by the operator or specific quality assurance personnel. Based on the quality, the operator decides whether to accept the current parameterization or continue the cycle. If the operator decides to continue the parameterization process, a new parameterization is chosen, which is suitable to address quality defects of the previously produced part. This workflow is repeated until a successful parameterization is chosen or other underlying hindrances, such as identified hardware problems, have to be addressed. In the latter case, the parameterization process needs to be restarted after underlying hindrances have been mitigated.

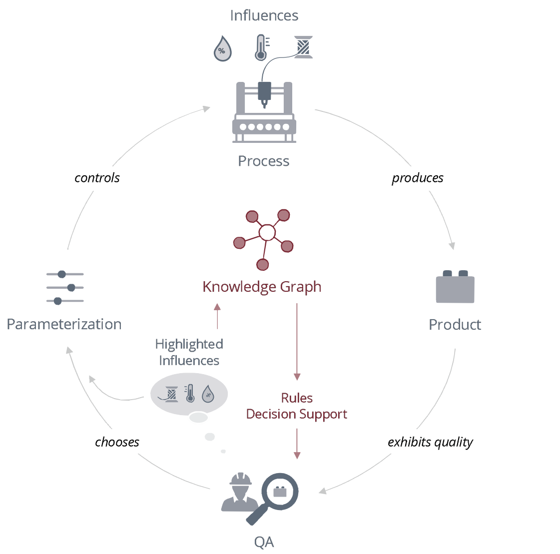

Fig. 1. Current parameterization workflow (grey) consisting of a process that is controlled by a parameterization, which is chosen iteratively by an operator to minimize quality defects of the produced part in regards to target criteria. Proposed integration (red) into the workflow to gather tacit process knowledge in an example-based manner.

Data-based approaches to facilitate parameterization have been previously proposed [2], [3]. Parameterization processes, however, are rarely executed during a machines lifetime and are hard to simulate due to complex physical interactions. Therefore, current supervised and reinforcement learning approaches are difficult to implement in practice. We envision an approach that includes semantic process knowledge represented in a knowledge graph to supplement the data that is being gathered. If the knowledge graph includes the experts' tacit knowledge of which factors influence their parameterization decision, we anticipate that the learning process can be guided accordingly [4]. To extract tacit knowledge, a process called knowledge extraction or elicitation needs to be performed by knowledge engineers and experts. Knowledge extraction of tacit knowledge constitutes the knowledge acquisition bottleneck [5] since, even with modern methods, it remains a time intensive process [6]. Also, difficulties appearing during knowledge extraction have to be addressed, such as the knowledge engineering paradox, which states that the more competent the experts, the less they are able to describe their solutions [7].

In this paper, we present an approach to gather semantic process knowledge and construct a corresponding knowledge graph, which includes tacit expert knowledge through crowdsourcing the experts during the manufacturing process. We show that information contained in the knowledge can be extracted into easily understandable rules. To facilitate the knowledge aggregation, we propose to integrate human-machine-interfaces (HMIs) to enable experts to provide insights into their reasoning as to why a certain parameterization is chosen. This has the benefit that vertices and edges in the knowledge graph automatically have a semantic that is directly related to the one present in the data, facilitating its application. Our approach has two benefits over existing work in knowledge extraction: (1) we assume that it is less intrusive than separate interviews, which are often a necessary part of expert knowledge extraction [6], leading to fewer requirements of specially trained personnel and (2) it is more time efficient since in many manufacturing processes experts have a lower workload between launching the process execution and evaluating the product. This time can be used by the experts to share their insights with our approach.

While the presented approach requires a certain amount of time, we assume that it is less intrusive. Furthermore, our approach addresses the knowledge engineering paradox, since it is example based, does not suffer from group dynamics that can occur if several experts are interviewed together, and is able to function without a knowledge engineer.

The remainder of this work is structured as follows: Section II gives an overview of related approaches to knowledge extraction. A formalization of the parameterization process and our methodology are presented in Section III. We describe the case study of fused-deposition-modeling on which we evaluate our approach in Section IV. The evaluation is presented in Section V. An outlook and a conclusion is given in Sections VI and VII, respectively.

II. Related Work

Knowledge extraction can be classified into two categories, human-centered and data-based knowledge extraction. In this section we will provide an overview over both and describe the special case of human-in-the-loop systems.

a) Human-Centered Knowledge Extraction: Traditionally, knowledge extraction or elicitation is performed by a knowledge engineer and one to several experts. The engineer chooses from an array of different techniques such as (semi-) structured or unstructured interviews, observations, protocol analysis or sorting and rating [8]. In the context of manufacturing scenarios, Deslanders et al. explored structured interviews to extract production rules for parameters that directly influence part quality [9]. Combinations of the above mentioned techniques have been successfully applied to the extraction of tacit procedural knowledge as mixed method approaches [6].

Seymoens et al. describe a different approach that aims to extract tacit knowledge by integrating domain experts during algorithm creation [10]. However, the experts' knowledge is only implicitly contained in the resulting algorithms. Therefore, it is not available explicitly for onboarding purposes or similar purposes.

Gebus et al. propose a factory-wide knowledge-based decision support system for remedying faults. They argue for an approach limiting the knowledge engineer's interactions with domain experts, instead enabling experts through systems that interview them on a case based basis [7]. Kharlamov et al. proposed ontologies to represent industrial information models in manufacturing scenarios. In Addition, they presented a tool which allows engineers to build these ontologies, without deep knowledge on semantic technologies or ontology creation [11]. We follow their approach to move ontology or knowledge formalization towards people that are closer to the process as opposed to external ontology engineers. However, our approach requires no previous knowledge of semantic technologies and is more suited for operators, which usually have a lower level of education compared to engineers.

A crowdsourcing approach to knowledge extraction and decentralized knowledge graph construction is pursued by Wang et al. [12]. By voting on proposed triples, coworkers create a knowledge graph of techniques and rate a worker's perceived technical skill for a certain technique. While we also pursue a crowdsourcing approach, we are focused on tacit procedural and conceptual knowledge rather than factual knowledge.

b) Data-Based Knowledge Extraction: Logical analysis of data can be used to extract if-then rules as shown for parameters and influences in a fault diagnosis scenario by Bai et al. [13].

In addition, several natural language processing [14]–[17] based approaches exist for various domains, such as oil and gas or health care [18]. A different approach is described by Katti, which allows ontologies to be generated from annotated source code of manufacturing execution systems [19]. However, both kinds of approaches are not directly applicable to our scenario since they are limited to factual knowledge. An approach that is specific to procedural knowledge is presented by Pareti et al. [20]. However, it would still require textual descriptions of the operators' thought processes during parameterization which rarely exists for manufacturing processes, possibly due to the knowledge engineering paradox.

c) Human-in-the-Loop Learning Systems: In human-in-the-loop or interactive learning systems, an expert gives feedback to the suitability of a learning system's prediction, see for example [21]. This, like our approach, constitutes a data-based knowledge extraction. However, this knowledge is usually fed directly into the learning system and therefore remains implicit as opposed to our approach which makes it explicit. Additionally, in our case, the externalized knowledge originates from the same operator that defined the specific parameterization, leading us to the assumption that it is of higher quality. Furthermore, interactive systems are at risk of influencing the expert by the data that is shown to them [22], which is not the case in our approach.

III. Methodology

In this section, we present a formalization of the parameterization process found in many manufacturing processes and introduce our approach to decentralized knowledge extraction.

A. *Formalizing Parameterization of Manufacturing Processes*

Our approach is suited to manufacturing processes where the quality of the produced part is influenced by a multitude of factors. This section provides a formalized view of the parameterization process practiced in such processes. Influential factors can be divided into factors the machine operator can adjust during production and those that are considered nonadjustable (e.g. environmental conditions). Adjustable parameters are referred to as process parameters, a concrete instance of which is referred to as a parameterization. Parameterization choices are made on the basis of nonadjustable factors $\mathcal{A} = \mathcal{O} \sqcup \mathcal{C} \sqcup \mathcal{T}$, where the multi-dimensional expressions of part or object characteristics $\mathcal{O}$, target criteria $\mathcal{C}$ and environmental factors $\mathcal{T}$ are dependent on the manufacturing process.

Based on our observations of parameterization processes in different manufacturing domains, we propose that operators choose the parameterization according to a complex internal mental model that is based on prior knowledge, education and experience:

$$p_i = f(\{o, c, \tau\}, \{p_0, \ldots, p_{i-1}\}, \{q_0, \ldots, q_{i-1}\}, \mathcal{M}) \qquad (1)$$

with:

- object characteristics $o \in \mathcal{O}$,
- target criteria $c \in \mathcal{C}$ (e.g. optimal quality or minimal cycle times),
- current environmental parameters $\tau \in \mathcal{T}$ that are dependent on the process e.g. environment temperature, material used or machine health,
- process parameters $p \in \mathcal{P}$ for previous iterations,
- quality $q \in \mathcal{Q}$ achieved at previous iterations under process parameters p and
- tacit declarative, procedural and conceptual knowledge available to the machine operator that has been gained through e.g. instruction, training or education $\mathcal{M}$.

$\mathcal{M}$ is assumed to be relatively constant during the optimization process, since the optimization process is of short duration compared to the time the operator has spent learning and building $\mathcal{M}$.

Initially, a production process is parameterized by a standard parameterization p_s of default values, which is, for example, determined during initial setup of the production process. To evaluate $p_0 = p_s$, the operators execute the manufacturing process resulting in a part with the quality q_0. Operators then try to iteratively—the current iteration is denoted as i—find a (pseudo-) optimal parameterization fulfilling the constraints given by $\{o, c, \tau\}$. Just as with the initial parameterization, the current parameterization p_i is evaluated by executing the manufacturing process and subsequently evaluating the quality of the produced part by quality assurance, leading to new q_i. Based on q_i and additional constraints, such as a set budget for parameterization, the operator decides whether to conclude the iterative process or to continue in hopes of finding a parameterization that returns a quality better fulfilling the target criteria. In the latter case, $f(\cdot)$ is evaluated anew. Usually, however, the iterative search concludes when a good parameterization is satisfactory in regard to the target criteria and can not be improved upon within a few iterations. After the iterative search has concluded, selecting $\hat{p}^* = \arg\max_p\{q_p\}$ for production completes the parameterization process and allows production to commence.

B. *Decentralized Knowledge Extraction*

To extract tacit procedural as well as conceptual knowledge of the operators' mental model $\mathcal{M}$, we propose a system based on recording individual parameterizations returned by $f(\cdot)$ and the influences used as inputs. For each evaluation of their respective function $f_{\mathcal{M}}(\cdot)$, operators are asked to provide the set

$$\alpha_i = \{x \mid x \in \mathcal{A} \sqcup \mathcal{P} \sqcup \mathcal{Q} \text{ and } \\ x \text{ influences the choice of values for } p_i)\}$$

of the most influential values from $f_{\mathcal{M}}$'s inputs for the parameterization p_i. These influences are selectable in an extension of their familiar human machine interface (HMI). Furthermore, they are asked to provide their confidence $\zeta_i \in [0,\ 1]$, that the chosen influences are correctly identified.

We then construct a knowledge graph k_i defined by the vertices $v_i = p_i \cup \alpha_i$ and the edges $e_i = p_i \times \alpha_i$. Note that, since the experts are not asked to establish direct links between individual influences and parameters, we are not able to differentiate which influences α resulted in which parameters p. Therefore, we have to treat all edges as a candidate for an influenced-by relation. This, however, decreases the task complexity for operators considerably, since they do not have to decide which parameter relations actually exist. Choosing a knowledge graph as a data structure to contain the information makes it easy to combine multiple knowledge graphs describing different aspects of the manufacturing process [23]. Since we assume that the parameterization process starts with a standard parameterization p_s, only the vertices and edges are included where p_i diverges from p_s. This leads to a considerably more compact representation of the relevant segments of the operators' knowledge. When generating the full knowledge graph by aggregating $k = \bigcup_i k_i$, we additionally weigh individual edges with ζ_i, where the final weight is the mean of all aggregated weights.

To lower the amount of parameterization iterations required until a satisfactory parameterization is achieved for a task $\omega = \{o, c, \tau\}$, we propose to integrate the information contained in the knowledge graph into a decision support system for part and target criteria specific parameterizations. If o and

c are fixed, we can analyze $(\alpha, \hat{p}^*, q)$ relationships for all encountered tasks $\omega \in \Omega$, where Ω encompasses all tasks and closely relates to $\mathcal{A}$ in that $\omega \subseteq \mathcal{A}$. In this work, we assume that $\hat{p}^*$ is not yet well known due to a strong limitation of executed parameterizations and thus operate on (α, p, q) triples for operator support.

A knowledge graph is bound by the task ω observed during knowledge extraction, in that it is only possible to interpret it reliably for a new ω if one has a notion of how it relates to the one observed originally. Given this information, parameterization suggestions p_{ω_i} can be calculated based on the similarity of ω_i, the task the operator is currently faced with, to previously encountered tasks, e.g. by taking the mean of good parameterizations for similar tasks. Similarity can either be directly calculated on characteristics of ω, or, if these are not observable in raw data, through cluster detection of (α, p, q) triples and an operator's input, which cluster might be best suited to ω_i, e.g. because it contains ωs with visually similar parts o.

IV. Case Study: Fused-Deposition-Modelling

To evaluate the proposed methodology, we apply our approach to plastics based Fused-Deposition-Modelling (FDM), an additive manufacturing technique and adjust the formalization accordingly. In FDM, parts, described by 3D CAD models, are printed layer-wise by a nozzle which melts and deposits raw material on a printbed [24]. To transform 3D models into specific instructions for the printer, a slicing software is used that generates necessary support structures, computes infill structures and calculates the tool path. Therefore, FDM contains not only process parameters pertaining to temperatures and motor actuation but also a multitude of higher-level process parameters that are configured in the slicing software and thereby have an indirect effect on instructions for the printer. Consequently, the process parameters for FDM are defined by the slicing software's parameters. Both slicing software and process parameters can slightly vary between machine manufacturers.

As a slicing software that acts as our HMI, we use Cura[1] which offers approx. 540 parameters for the printers we use, of which approx. 50 are regularly adjusted by operators, e.g. *print_bed_temperature* and *material_print_temperature*. We treat the respective printers' default configuration in Cura as the default parameterization p_s as those are usually derived from at least some experience and constitute a good starting point for operators. Objects $\mathcal{O}$ consist of all 3D model files, with dimensions fitting the printers' capacity. $\mathcal{Q}$ is defined by eleven visually detectable quality metrics such as warping or stringing. Quality assurance is carried out by at least one trained operator after a part is produced. The operator rates up to eleven quality characteristics per part on a discrete scale of 0 to 5 using a web-frontend.

While a number of target criteria can be formulated, for this case study, we limit them to the optimal quality in regards to metrics contained in $\mathcal{Q}$. Since c is constant, we can simplify $f(\cdot)$ accordingly. While environmental factors $\mathcal{T}$ can be plentiful, for this case study, we narrow them down to: humidity, temperature, printer and the raw material, separated by producer, type and color. We do not treat $\mathcal{P}$ as an influential factor since dependencies between process parameters are automatically calculated by Cura.

Our experimental evaluation is therefore based on

$$p_i = f^{\mathrm{FDM}}(\{o, \tau\}, \{p_s, \dots, p_{i-1}\}, \{q_s, \dots, q_{i-1}\}, \mathcal{M}) \quad (2)$$

and

$$\alpha_i^{\mathrm{FDM}} = \{x \mid x \in \mathcal{A}^{\mathrm{FDM}} \sqcup \mathcal{Q} \text{ and } x \text{ influences the choice of values for } p_i)\},$$

with $\mathcal{A}^{\mathrm{FDM}} = \mathcal{O} \sqcup \mathcal{T}$. We equipped the printers with humidity and temperature sensors and extended Cura's user interface via a plugin. Figure 2 shows the extended user interface. After process parameters have been adjusted in the right-side window, the operators are now able to view and select influential environmental parameters, quality characteristics of previously produced parts in the dialog shown on the left. Also, they are able to share insights via a text field and give an indication of their confidence for the chosen parameterization.

The information gained through the HMI is aggregated as described in Section III and can be utilized as part of a decision support system. Based on the knowledge graph, the operator is able to do two things: Firstly, given the current task ω_i, the operator is able to check against known connections, i.e. edges in the graph, between Ω and process parameters $\mathcal{P}$, which gives him an overview of a suitable set of process parameters to adjust. By selecting an edge, the operator is able to visualize the underlying data. After ascertaining the degree of similarity between ω_i and previously recorded ones, similarities in recorded parameterizations are visible. Secondly, if confronted with a specific problem regarding a quality characteristic $q_{i-1,j}, j \in \{1, \dots, 11\}$, the operator is able to filter the graph for process parameters which are influenced by the same quality characteristic, thus, showing recorded changes in parameterization able to remedy this specific quality deficit.

A. Dataset

The dataset we use for evaluation, was intermittently gathered by five operators of differing experience levels on three machines over the course of several months. Before the data was recorded, operators were instructed with the target criteria. They were then given the respective 3D object file and information about the environmental factors, as well as access to the HMI.

The dataset contains 65 examples distributed over 25 parameterization processes. The mean of iterations per parameterization was 2.6 with a standard deviation of 3.2619. 17 processes contain only a single example, thus, implying that the initially derived configuration was creating a part of sufficient quality. In contrast, the longest parameterization process took 15 iterations.

[1] https://ultimaker.com/software/ultimaker-cura

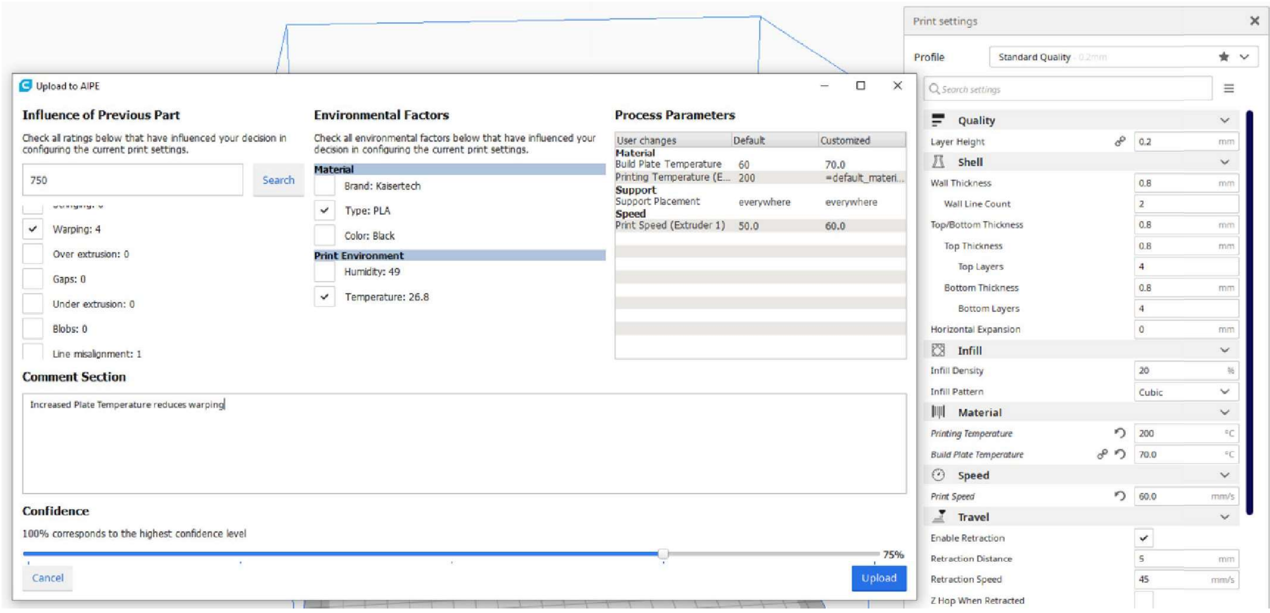

Fig. 2. Extended Cura HMI. First, a parameterization is defined in the *Print settings* dialog to the right. Then, influential quality characteristics and environmental factors can be highlighted (center left) and the part is produced.

Operators usually varied zero to four parameters with the majority being either a single parameter or three. The number of parameters adjusted declined over the parameterization process, implying that seemingly optimal configurations for some were found while others had to be additionally fine tuned for the task at hand. The most commonly named influential factors pertained to the material or previous ratings. Interestingly, operators seemed not to be able to discern whether producer, chemical mixture (and thus the colour) or the raw material, was important and tended to ascribe changes to all three. Typically, one to three influences from a previous rating, describing the critical shortcomings of the part, were deemed influential. Rating categories where the rating was optimal, were not named as influential, thus operators seem to be confident that those will not suffer from the change at hand. Operator confidence for parameter choices varied during the process. Both 0% and 100% were named. Interestingly, in one instance a 100% confidence in the identified influential factors was still followed by four additional parameterization tests with decreasing confidences (75%, 50%, 25% and 25%, respectively). We assume that there could be two possible explanations for such a pattern: (1) initial overconfidence which was lowered after the parameterization did not have the desired effect or (2) after achieving a certain quality, the operator tries out different parameterizations to see if the quality can be boosted further in a trial-and-error manner. In one instance, where the initial parameterization had a confidence of 0%, the operator showed 100% confidence in his attribution of influences for the last iteration and was able to conclude the process.

V. Experimental Evaluation

To evaluate whether our approach constitutes a benefit to knowledge extraction, we conducted two experiments. The first assesses the effect of the influences provided by the experts compared to a purely data driven knowledge graph aggregation approach based on all available factors. The second evaluates whether it is possible to extract rules from the knowledge graph and accompanying influence visualizations. Both experiments were conducted on knowledge graphs aggregated on the dataset described in Section IV. Direct comparison to other approaches is compounded by two factors: (1) to our knowledge there exist no directly comparable algorithmic approaches and (2) our approach requires a specific data structure (the influential parameters highlighted by the operator) that are absent from any public dataset we know of.

A. Effect of Provided Influences

To evaluate whether the influences α, that were highlighted by the experts, offer a benefit, we compare the knowledge graph based on the experts' opinion of influential parameters KG_α (cf. Figure 3) to a knowledge graph that takes every observed combination of α and adjusted process parameters $\{p_x \,|\, p_{x,i} \in p_x$ and $p_{x,i}$ differs in value value in from $p_{s,i}\}$ into account $KG_{A \sqcup Q}$ (cf. Figure 4). Comparing their respective visualizations, it is directly discernible that $KG_{A \sqcup Q}$ may theoretically contain knowledge, however, it is not obvious without tedious inspection of all 260 edges. In contrast, KG_α constitutes a dimensionality reduction, leading to a concise overview of which parameters are dependent on which influences. Looking at the amount of vertices and edges this is evident as well. While $KG_{A \sqcup Q}$ contains 33 vertices

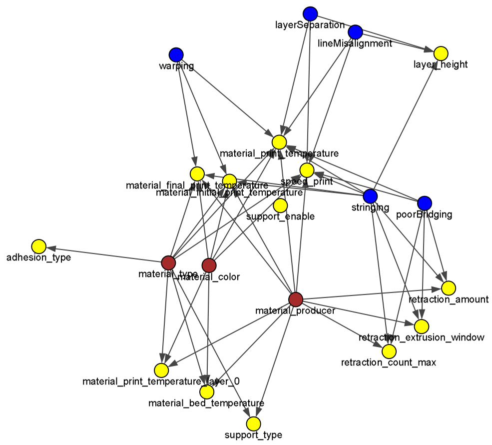

Fig. 3. Knowledge graph KG_{α}, constructed taking into account only influences marked by experts. Yellow, brown and blue nodes are process parameters p, environmental factors and influential quality characteristics, respectively.

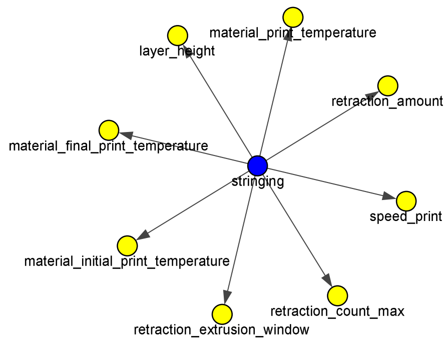

Fig. 5. Knowledge graph KG_{α}, filtered for influential quality characteristic stringing. Yellow and blue nodes are process parameters p and influential quality characteristics, respectively.

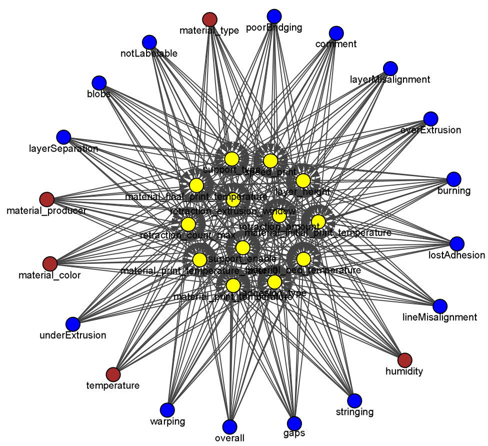

Fig. 4. Knowledge graph $KG_{A \sqcup Q}$, constructed taking all influences into account. Yellow, brown and blue nodes are process parameters p, environmental factors and influential quality characteristics, respectively.

and 260 edges, KG_{α} contains 21 vertices and 46 edges. Overall, investigating 46 edges, some of which can be instantly discarded because they do not contain similar data to ω, is a task that can be performed by a non-expert to gain a highlevel understanding of the FDM process.

B. Evaluation of Data in Knowledge Graph

To evaluate whether we are able to extract meaningful rules from the knowledge graph, we compare them to rules existing in a natural language knowledge base for FDM. For Simplify3D [25], an alternative slicing software, such a knowledge base exists. It contains rules that are based on practical experiences, in depth know-how of slicer functionality as well as process knowledge regarding physical relationships. Exemplarily, we focus on rules pertaining to a quality characteristic called stringing, which is visible through fine strands of filament erroneously connecting pieces of the part. We manually extracted all corresponding rules out of Simplify3D's knowledge base, selected the ones pertaining to process parameters and transferred Simplify3D-specific settings to their Cura equivalent. This results in the following rule set:

1) If stringing is present, then increase *retraction_amount* in 1mm increments up to 15mm or until stringing is not present anymore
2) If stringing is present, then adjust *retraction_speed* between 20 and 100mm/s
3) If stringing is present, then lower *material_print_temperature* by 5 to 10°C
4) If stringing is present, then set *combing* to true
5) If stringing is present, then increase *speed_print*

To extract rules out of the knowledge graph, we first filter it by the influential quality characteristic stringing, which results in the subgraph displaying the process parameters that have been adjusted to minimize stringing. The result is shown in Figure 5. After filtering, we have eight process parameters that are candidates influencing stringing. They were collected through a total of six tasks for which eleven parameterization iterations were completed. To visualize the relationship between the influential quality characteristic and the process parameters, we plotted each (α, p, q) triple relatively, i.e. relative quality improvements $\alpha - q$ on the y axis and relative parameter changes $p - p_s$ on the x axis, respectively. An example is shown in Figure 6, where *retraction_amount* changes are plotted against improvements in stringing quality.

Analyzing similar visualizations for all parameter candidates led to the following rules:

1) If stringing is present, then reduce *retraction_extrusion_window* by up to 3mm
2) If stringing is present, then increase *retraction_count_max* by 50
3) If stringing is present, then increase *retraction_amount* by 2mm
4) If stringing is present, then moderately decreasing *speed_print* could have a positive effect
5) If stringing is present, then increasing *material_print_temperature* could have a positive effect

layer_height is not present in the rule set since it seems to be an outlier, either caused by accidental incorrect operation or an incorrect understanding of the HMI.

Comparing the results at an influence level, we observe that there is an overlap of 37.5% regarding influential factors between Simplify3D's knowledge base and our knowledge graph. Furthermore, 60% of Simplify3D's influential factors were also present in the knowledge graph. Comparing the results at a rule level, we can see an overlap of 14.28% in recommendations between Simplify3D's and our rule base. We agree with 20% of Simplify3D's rules, disagree with 40% and do not cover an additional 40%. For the contradicting rules, our value recommendations are based on contradictory data present in the iterations. This implies a high level of uncertainty and leads to their cautious phrasing. Possible reasons for the contradictory data are the parallel adjustment of other process parameters, printer or material characteristics, or erroneous information provided by the experts. However, our approach discovered two rules that are not present in Simplify3D's knowledge base. This might be the case since the knowledge could be specific to Cura and not be directly transferable to Simplify3D. Additionally, they might be very specific to both the printers used as well as the environment in which production took place, while the Simplify3D guide obviously aims at great generality. Our rules were generated based on data of only eleven iterations and might not be representative for all cases, as a multitude of other factors could have affected the production process apart from the one process parameter analysed for each rule. We assume that with more samples, the results will exhibit a greater level of confidence. However, especially at the influence level, they can be considered satisfactory. The effectiveness regarding the extra time spent by operators cannot be compared between the two approaches since Simplify3D provides no metrics regarding time spent on creating their knowledge base. Since our approach is very effective in regards to the amount of data required, we are inclined to view the cost/benefit ratio as positive.

VI. Outlook

We observed different patterns in the data we gathered (cf. Section IV). Since this increases the difficulty of interpreting the results in a standardized way, we plan to enhance the HMI to motivate a more consistent operator behavior. If the

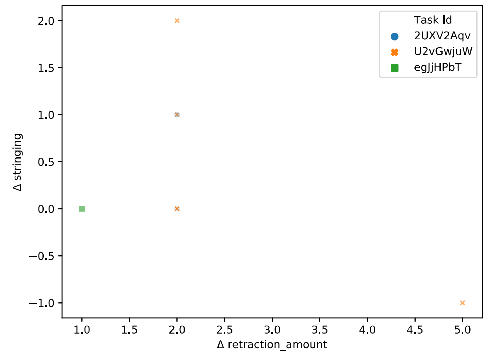

Fig. 6. Changes in *retraction_amount* (in mm) relative to the default parameterization p_s and corresponding relative changes in quality (higher is better, 5 is the maximum) to the iteration that was selected as influential.

operators are unable to highlight influences, the data is likely to be negatively affected. Consequently, we provided a text input as fallback. Analyzing the entries with natural language processing methods, applying ontology learning methodologies and fusing the results with the knowledge graph created based on the directly selectable influences is likely to be a key factor to increasing robustness of the approach. Furthermore, more advanced weighting strategies have to be evaluated to better filter out outliers or erroneous information.

For potential industrial applications, it is necessary to establish trustworthiness of experts or the information they provide. Apart from technical aspects that have to be researched in that regard, the resulting ethical aspects have to be addressed. To increase usability of the approach as a decision support system, similarity metrics for o, i.e. parts that should be produced, have to be researched. Also, automatically extracting rules based on relative changes of (α, p, q) triples would constitute a benefit in usability. Transferring the presented approach to different domains and validating its assumed domain independence at an abstract level as well as a broader field study, which validates the operators' acceptance and the compatibility of the presented approach with their specific workflows, is planned. We also intend to conduct this study on a process where traditional ontology extraction methodology has already been applied. Therefore, it would be particularly interesting to compare results after a certain data foundation has been gathered quantifying the respective effectiveness.

Integrating the proposed approach with knowledge graphs describing manufacturing systems at a different level of abstraction and therefore containing knowledge of a different kind also presents itself as an interesting research field. Differences in semantic notation are bound to appear in this case, placing a focus on the alignment of knowledge graphs.

Additionally, we want to investigate whether knowledge collected with the presented approach can be utilized to increase performance of supervised learning systems through the ability to present not only label but also additional information pertaining to the decision processes involved [4].

VII. Conclusion

In this paper, we formalized the parameterization process found in many manufacturing scenarios. Building on that, we presented an abstract methodology for knowledge extraction: while actively parameterizing machinery in a production environment, operators highlight exemplary values that influenced their decision making. The methodology is designed to be able to be applied during production without additional personnel and without causing delays. We described a decision support system building on the knowledge graph constructed by the proposed methodology. This can be used to decrease onboarding times as well as provide assistance when confronted with a concrete problem.

Illustrating our approach, we conducted a case study demonstrating that the abstract methodology can be applied to a fused-deposition-modeling (FDM) process. In the domain of FDM, we collected a real-world dataset on which we conducted an experimental evaluation of the proposed approach. We showed that 20% of rules existing in a third-party knowledge base regarding a specific quality characteristic can be approximated with the help of our approach requiring only eleven examples pertaining to this specific quality characteristic. Therefore, we are confident that the described methodology can form the foundation for the application of a combination between knowledge based and learning systems in manufacturing.

Acknowledgment

This work is supported by the German Federal Ministry for Economic Affairs and Energy (BMWi).

References

[1] E. Permin, F. Bertelsmeier, M. Blum, J. Bützler, S. Haag, S. Kuz, D. Özdemir, S. Stemmler, U. Thombansen, R. Schmitt, C. Brecher, C. Schlick, D. Abel, R. Poprawe, P. Loosen, W. Schulz, and G. Schuh, "Self-optimizing Production Systems," *Procedia CIRP*, vol. 41, pp. 417–422, 12 2016.

[2] E. Lughofer, C. Zavoianu, R. Pollak, M. Pratama, P. Meyer-Heye, H. Zörrer, C. Eitzinger, and T. Radauer, "Autonomous Supervision and Optimization of Product Quality in a Multi-stage Manufacturing Process based on Self-adaptive Prediction Models," *Journal of Process Control*, vol. 76, pp. 27–45, 04 2019.

[3] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Rewards," 2019.

[4] R. Nordsieck and J. Hähner, "Interactive knowledge-guided learning," in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. IEEE, 2020, pp. 237–239.

[5] M. L. Shaw and B. R. Gaines, "An interactive knowledge-elicitation technique using personal construct technology," in *Knowledge acquisition for expert systems*. Springer, 1987, pp. 109–136.

[6] L. Hörner, M. Schamberger, and F. Bodendorf, "Externalisierung von prozess-spezifischem mitarbeiterwissen im produktionsumfeld," *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, vol. 115, no. 6, pp. 413–417, 2020.

[7] S. Gebus and K. Leiviskä, "Defect-related knowledge acquisition for decision support systems in electronics assembly." in *ICINCO-ICSO*, 2007, pp. 270–275.

[8] N. Shadbolt, P. R. Smart, J. Wilson, and S. Sharples, "Knowledge elicitation," *Evaluation of human work*, pp. 163–200, 2015.

[9] V. Deslandres and H. Pierreval, "Knowledge acquisition issues in the design of decision support systems in quality control," *European journal of operational research*, vol. 103, no. 2, pp. 296–311, 1997.

[10] T. Seymoens, F. Ongenae, A. Jacobs, S. Verstichel, and A. Ackaert, "A methodology to involve domain experts and machine learning techniques in the design of human-centered algorithms," in *IFIP Working Conference on Human Work Interaction Design*. Springer, 2018, pp. 200–214.

[11] E. Kharlamov, B. C. Grau, E. Jiménez-Ruiz, S. Lamparter, G. Mehdi, M. Ringsquandl, Y. Nenov, S. Grimm, M. Roshchin, and I. Horrocks, "Capturing industrial information models with ontologies and constraints," in *International Semantic Web Conference*. Springer, 2016, pp. 325–343.

[12] S. Wang, C. Huang, J. Li, Y. Yuan, and F.-Y. Wang, "Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts," *IEEE Access*, vol. 7, pp. 136 951–136 961, 2019.

[13] X. Bai, J. Tan, and X. Wang, "Fault diagnosis and knowledge extraction using fast logical analysis of data with multiple rules discovery ability," in *International Conference on Intelligence Science*. Springer, 2018, pp. 412–421.

[14] T. R. Bandaragoda, D. De Silva, D. Alahakoon, W. Ranasinghe, and D. Bolton, "Text mining for personalized knowledge extraction from online support groups," *Journal of the Association for Information Science and Technology*, vol. 69, no. 12, pp. 1446–1459, 2018.

[15] P. Schumacher, M. Minor, K. Walter, and R. Bergmann, "Extraction of procedural knowledge from the web: A comparison of two workflow extraction approaches," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 739–747.

[16] A. Gangemi, "A comparison of knowledge extraction tools for the semantic web," in *Extended semantic web conference*. Springer, 2013, pp. 351–366.

[17] M. Mohamed, S. Pillutla, and S. Tomasi, "Extraction of knowledge from open government data," *VINE Journal of Information and Knowledge Management Systems*, 2020.

[18] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, and H. M. Abbasi, "A survey of ontology learning techniques and applications," *Database*, vol. 1, p. 24, 2018.

[19] B. Katti, "Ontology-based approach to decentralized production control in the context of cloud manufacturing execution systems," Ph.D. dissertation, TU Kaiserslautern, 2020.

[20] P. Pareti, B. Testu, R. Ichise, E. Klein, and A. Barker, "Integrating know-how into the linked data cloud," in *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 2014, pp. 385–396.

[21] I. Sundin, T. Peltola, L. Micallef, H. Afrabandpey, M. Soare, M. Mamun Majumder, P. Daee, C. He, B. Serim, A. Havulinna *et al.*, "Improving genomics-based predictions for precision medicine through active elicitation of expert knowledge," *Bioinformatics*, vol. 34, no. 13, pp. i395–i403, 2018.

[22] P. Daee, T. Peltola, A. Vehtari, and S. Kaski, "User modelling for avoiding overfitting in interactive knowledge elicitation for prediction," in *23rd International Conference on Intelligent User Interfaces*, 2018, pp. 305–310.

[23] X. Wilcke, P. Bloem, and V. De Boer, "The knowledge graph as the default data model for learning on heterogeneous knowledge," *Data Science*, vol. 1, no. 1-2, pp. 39–57, 2017.

[24] I. Gibson, D. Rosen, and B. Stucker, *Additive Manufacturing Technologies - 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer-Verlag, 2015.

[25] Simplify3D, "Stringing or oozing," 2020, visited on Oct. 25, 2020. [Online]. Available: https://www.simplify3d.com/support/print-quality-troubleshooting/stringing-or-oozing/