Knowledge Graph Supported Machine Parameterization for the Injection Moulding Industry

Stefan Bachhofner[1(✉)], Kabul Kurniawan[1,2], Elmar Kiesling[1], Kate Revoredo[1], and Dina Bayomie[1]

[1] Institute for Data, Process and Knowledge Management, Vienna University of Economics and Business, Vienna, Austria
{stefan.bachhofner,kabul.kurniawan,elmar.kiesling,kate.revoredo, dina.bayomie}@wu.ac.at

[2] Austrian Center for Digital Production, Vienna, Austria

Abstract. Plastic injection moulding requires careful management of machine parameters to achieve consistently high product quality. To avoid quality issues and minimize productivity losses, initial setup as well as continuous adjustment of these parameters during production are critical. Stakeholders involved in the parameterization rely on experience, extensive documentation in guidelines and Failure Mode and Effects Analysis (FMEA) documents, as well as a wealth of sensor data to inform their decisions. This disparate, heterogeneous, and largely unstructured collection of information sources is difficult to manage across systems and stakeholders, and results in tedious processes. This limits the potential for knowledge transfer, reuse, and automated learning. To address this challenge, we introduce a knowledge graph that supports injection technicians in complex setup and adjustment tasks. We motivate and validate our approach with a machine parameter recommendation use case provided by a leading supplier in the automotive industry. To support this use case, we created ontologies for the representation of parameter adjustment protocols and FMEAs, and developed extraction components using these ontologies to populate the knowledge graph from documents. The artifacts created are part of a process-aware information system that will be deployed within a European project at multiple use case partners. Our ontologies are available at https://short.wu.ac.at/FMEA-AP, and the software at https://short.wu.ac.at/KGSWC2022.

Keywords: Semantic web · Knowledge graphs · Manufacturing process · Automotive industry · Failure mode and error analysis · Industry 4.0

This research has received funding from the Teaming.AI project, which is part of the European Union's Horizon 2020 research and innovation program under grant agreement No. 957402.

B. Villazón-Terrazas et al. (Eds.): KGSWC 2022, CCIS 1686, pp. 106–120, 2022.
https://doi.org/10.1007/978-3-031-21422-6_8

1 Introduction

In the manufacturing industry, the rise of the I4.0 paradigm facilitates complex data-driven use cases [12,13]. This has, however, only increased the need for deep domain knowledge to make sense of increasingly abundant data. In this context, Knowledge Graphs (KGs) have emerged as an important tool that is increasingly being adopted in manufacturing applications [5]. KGs can integrate data from a variety of sources and evolve their schema to accommodate growing requirements. Due to these properties, they have been used in industrial settings as a backbone for a variety of downstream tasks such as building digital twins, risk management, process monitoring, machine service operations, and factory monitoring [9]. Furthermore, they increasingly provide a foundation for machine learning and AI-driven applications in enterprise settings in general [2] and manufacturing in particular [27].

This opens up interesting opportunities in quality management. In this paper, we focus on the automotive industry and its supplier networks, in which the management of product quality along the production chain is crucial and the subject of various standards such as ISO/TS 16949:2009 [10]. In the injection moulding industry in particular – which supplies plastic parts to automotive manufacturers – the parameterization of production machines to achieve consistent output is a complex and delicate process that requires substantial domain knowledge [4]. Part of this domain-knowledge is codified in *injection process adjustment protocols*, which our industrial partner has integrated into their quality management processes (cf. Fig. 1). The main objective of such protocols is to document knowledge gained from (often long-term) experience and make it available in digestible form. In addition, Failure Mode and Effects Analysis (FMEA) documents are extensively used to describe those failure modes together with their potential causes and effects.

These documents are important tools to reduce the time machines spend in an unproductive state. In addition, they are used extensively as training materials for employees. However, both the adjustment protocol and FMEA are currently typically maintained in numerous spreadsheets and accompanied by *parameter sheets* in which injection technicians record parameter changes. This document-centric workflow makes it difficult for injection technicians to identify the root cause of product defects as well as to compare and link deviations in sensor measurements to parameter changes made by the injection technician according to the adjustment protocol. In addition, due to misspellings, lack of time, or an incentive mismatch, the recorded changes in the protocol are often plagued by data quality issues. Furthermore, the document-centric approach makes it difficult to operationalize unstructured FMEAs knowledge, for example relating it to issues on the shop floor to derive insights on how to resolve them.

In this paper, we address this challenge and contribute towards the vision of KG-based shop-floor support; more specifically, we propose an approach to enhance quality management on the shop floor that will serve as the backbone for various Artificial Intelligence (AI) applications within a larger process-aware Information System (IS). We integrate *injection process adjustment protocols* and

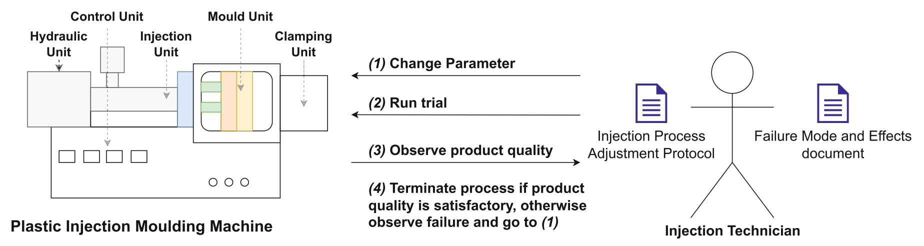

Fig. 1. Sketch of the current parameter adjustment process (plastic injection moulding machine adapted from [4]). Figure 2 shows the process formalized with Business Process Model and Notation (BPMN).

FMEAs within a KG and provide pipelines to iteratively update this knowledge graph from the respective documents. Furthermore, we replace the parameter spreadsheets with a KG that directly receives the changes from the injection moulding machine. To this end, we introduce an *FMEA-injection process adjustment protocol* ontology that extends an existing FMEA ontology [20]. In addition, we provide a software library that transforms spreadsheets into a KG representation. This lowers the entry barrier towards integrating the KG-based approach into the current workflow. We motivate and validate the approach with an application in injection moulding, but expect that our artifacts - the ontologies and the tool - are applicable more generally in other production settings with similar requirements.

The remainder of this paper is structured as follows. Section 2 describes the problem based on the current workflow; Sect. 3 introduces the adjustment protocol ontology, and the adapted FMEA ontology, which we develop based on the documents provided by our industrial partner. Furthermore, this section describes the accompanying software and provides summary statistics on the constructed KG. In Sect. 4, we describe the application of the constructed KG as the backbone for a parameter recommendation system. Section 5 discusses the broader context of the process-aware IS that the KG and components introduced in this paper are part of, outlines empirical evaluation strategies, future work, and limitations. In Sect. 6, we review related work before concluding the paper in Sect. 7.

2 Problem Statement

In this section, we describe the current workflow of the injection moulding processes and outline the research problem. This includes a description of the production process and how adjustments to the injection moulding machine are managed.

Our use case stems from a large European automotive supplier that produces plastic car parts through injection moulding. To develop the use case, we conducted a domain analysis and modeled the current processes in Business Process Model and Notation (BPMN) [19]. The injection moulding process itself is complex – due to space constraints, we describe only the high-level stages, which

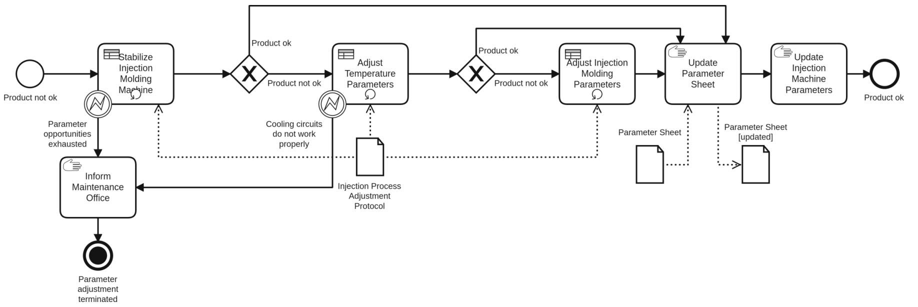

Fig. 2. Injection process adjustment protocol process in a BPMN model. The tasks of the process rely on documents, paper or digital, as an input or output.

are (cf. [4]): *(i) Filling*, i.e., feeding and melting the materials and injecting them into the mold; *(ii) Packing*, i.e., the pressing process to ensure the densely textured product is produced; *(iii) Cooling* until the molten material is fully solidified; and *(iv) Ejection*, i.e., the mold releasing process after the work piece has cooled to a given temperature. After these steps, the operator performs a quality inspection of the product and gives feedback on whether the product is *OK* or *Not OK (NOK)*. If the product is *NOK*, then the operator documents among other data points the type of defects and the condition under which the defect appeared. Based on these observations, the operator can adjust the parameters of the injection moulding machine according to specified procedures to address the issue.

The company manages these procedure during production with an *injection process adjustment protocol* (see Fig. 2). The protocol has three phases with *(i)* machine stabilization, *(ii)* temperature adjustment, and *(iii)* injection parameter adjustment. Within these phases, the protocol defines a sequence of actions that need to be taken to adjust a set of parameters; each action is associated with three aspects, which are *(i)* a parameter priority list, *(ii)* the direction in which the parameter should be changed, *(iii)* and the rate of change. Machine operators use this as a guiding tool, and it is supposed to shorten the time a machine spends in an unproductive state. The protocol requires the operator to manually readjust parameter values on the machine and update the parameter changes into a parameter sheet, which can be a source of error. Furthermore, the adjustment activities are carried out manually and are only supported by the guideline, which is provided in a text processing or spreadsheet format, without an automated feedback loop to the guideline. What is more, an FMEA document already exists that is supposed to be used to map defects to their specific causes and effects. This document is also in spreadsheet format, which makes it difficult to detect deviations from the guideline, identify the root cause of a given defect, and assess to what extent they are helpful, incorrect, and complete. Data quality is another key issue in this context, as it requires the shop floor employee to manually justify the reason for the deviation, which is error-prone and time-intensive. Another obstacle is that an ad-hoc analysis is expensive, and the absence of an immediate incentive for the operator to invest an effort in accurately describing a deviation.

3 Knowledge Graph for Product Quality Management

This section describes the knowledge sources, the developed FMEA-IPAP ontology which integrates the knowledge sources, the software components for automated extraction and construction from documents, and provides details and statistics on the constructed knowledge graph.

3.1 Knowledge Sources

The parameter adjustment knowledge graph integrates two major sources of knowledge with *(i)* the Failure Mode and Effects Analysis (FMEA) documents, and *(ii)* the Injection Process Adjustment Protocol (IPAP). We describe them next in greater detail.

Failure Mode and Effects Analysis (FMEA) is an engineering technique to define, review and identify potential failures and their effects and causes for systems, designs, processes or services [23]. The technique aims to describe, model, and analyze potential failures in order to ultimately improve product quality, increase productivity and reduce waste. This quality management tool has been widely used in many industrial applications and engineering domains. However, it is mostly conceived as a *"boring and complicated human activity"*, given its perception as complying with engineering regulations rather than improving product quality [24].

Injection Process Adjustment Protocols (IPAPs) contain a set of standard procedure definitions for injection parameter adjustment. Injection technicians use them, for instance, to adjust machine stabilization and mass temperature verification as well as parameter modification, where, depending on the type of failure mode (e.g., gloss or marbling) different sets of parameters such as compaction time or pressure, cooling time, and injection speed or pressure are changed individually and iteratively at a fixed rate until the production defect has been resolved. This protocol also includes information on the order in which parameters should be adjusted - the action priority. Action priority is a good example for knowledge derived over a long time period.

3.2 FMEA-IPAP Ontology

To represent concepts from both sources - described above - in an integrated *FMEA-IPAP ontology*, we developed *(i)* a parameter adjustment protocol ontology based on guideline documents used by the industrial partner, and *(ii)* an FMEA ontology based on an existing ontology and the FMEA documents also used by our industrial partner.

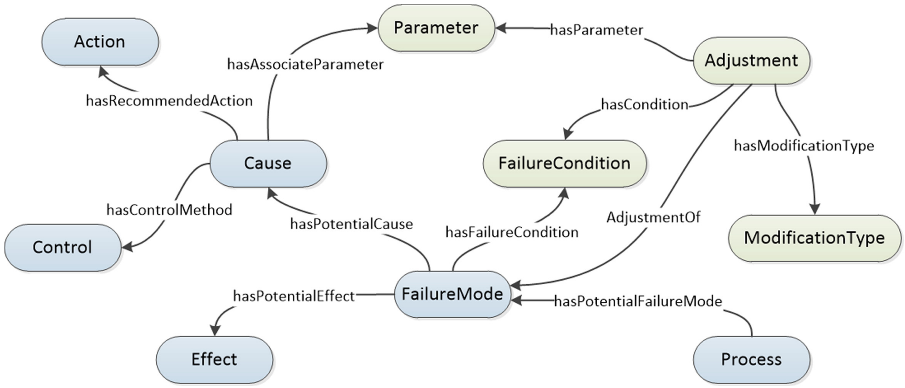

Fig. 3. FMEA Ontology (light blue) and IPAP Ontology (light green). Our FMEA ontology is adapted from [20]. (Color figure online)

To model the domain of interest, we started with a domain analysis that involved workshops and a review of the existing resources for FMEA and IPAP documents obtained from our industrial partner. Next, we conducted a survey of FMEA ontologies and identified the FMEA ontology proposed in [20] as a good candidate for reuse, due to the straightforward conceptual alignment with existing FMEA documents. For the representation of the knowledge from the IPAP documents, we did not find any suitable formalization in the literature and consequently used a bottom-up ontology construction approach [18] to create it.

As shown in Fig. 3, we combine the FMEA and IPAP concepts into a single integrated FMEA-IPAP OWL ontology[1]. The FMEA concepts (light blue) are partially adapted from an existing FMEA ontology proposed in [20]. The FMEA ontology consists of six main classes with seven object properties to link them. The FailureMode class defines the individual failure modes that may happen, such as *Marbling, Gloss, and Burn*. Each failure mode may be associated with the property `hasPotentialCause` to a number of potential causes (defined as Cause). Each Cause can have a recommended Action (linked via `hasRecommendedAction`). The IPAP ontology (light green) consists of four main classes, and four object properties to relate the instances of the classes. We reused several data properties from the Dublin Core Ontology[2] – e.g., `title` and `description` – to describe class details. Adjustment is the main class of the ontology that represents the individual adjustment protocol. It has data properties such as `actionPriority` that define the priority level of the adjustment (e.g. first, second, and third), and `adjustmentRate` to define the rate of adjustment. Furthermore, the Adjustment class has object properties to the classes *(i)* `hasParameter` links to Parameter, a class that describes specific parameters that need to be adjusted, such as *injection speed, mass temperature,*

[1] https://w3id.org/teamingai/resources/ont/FMEA.
[2] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.

and decompression, *(ii)* `hasCondition` provides a temporal and spatial context – i.e., under what conditions (FailureCondition) an adjustment is appropriate, for example *in the beginning, middle, last injection* (temporal aspects), *partially*, and *whole* (spatial aspects). *(iii)* `hasModificationType` to connect the ModificationType class that represents the type of modification (i.e., *decrease* and *increase*) The Adjustment class also links to classes in the FMEA ontology – the FailureMode class that matches the adjustment protocol (via `hasAdjustment` property and its inverse `adjustmentOf`) to the specific failure mode in the FMEA (e.g., marbling and gloss).

3.3 Construction Pipeline Implementation

We developed an extraction and transformation pipeline. The pipeline extracts the parameter adjustment rules and FMEA statements from spreadsheets, and then transforms them into a KG using the ontologies introduced in the previous section. We found that this automation dramatically reduces the entry barrier for the various stakeholders at our industrial partner. Moreover, it enables users unfamiliar with knowledge graphs to profit from our approach while they can use their familiar tool chain. To manage the creation of and interaction with the KG, and integrate it into existing workflows, we developed an Application Programming Interface (API) that provides three main functions – one function each for the transformations, and one function to recommend parameters. We designed the transformation functions to have the same function signature as they have the same responsibility, but for different ontologies. Function *recommend* is responsible for recommending a parameter adjustment action given a failure that arose under a condition. We refer the interested reader to the repository for further details.

3.4 Knowledge Graph Instance and Statistics

Figure 4 shows an excerpt of the knowledge graph output generated from both FMEA and Injection Process Adjustment Protocol (IPAP) documents. The two sources of knowledge are now integrated and linked. For example, a failure mode MARBLING links to potential cause Excessive Injection Speed that has an associated parameter Injection Speed; this parameter information has not been provided previously in the FMEA documents, but it is now being linked to the IPAP knowledge. The case is similar for the IPAP document, previously it had no information about the failure cause, but after integrating them into FMEA knowledge, we are able to directly trace the root cause of the failure – Excessive Injection Speed. The failure mode is also linked to the condition during injection process, which is the same entity as defined in the IPAP. We discuss the benefits of this integration and linking further in Sect. 4. Table 1 shows statistics of the developed ontology and the generated knowledge graph from both FMEA and IPAP. While they might evolve, we consider them as a static part, as the ontologies would change less compared to their instance data.

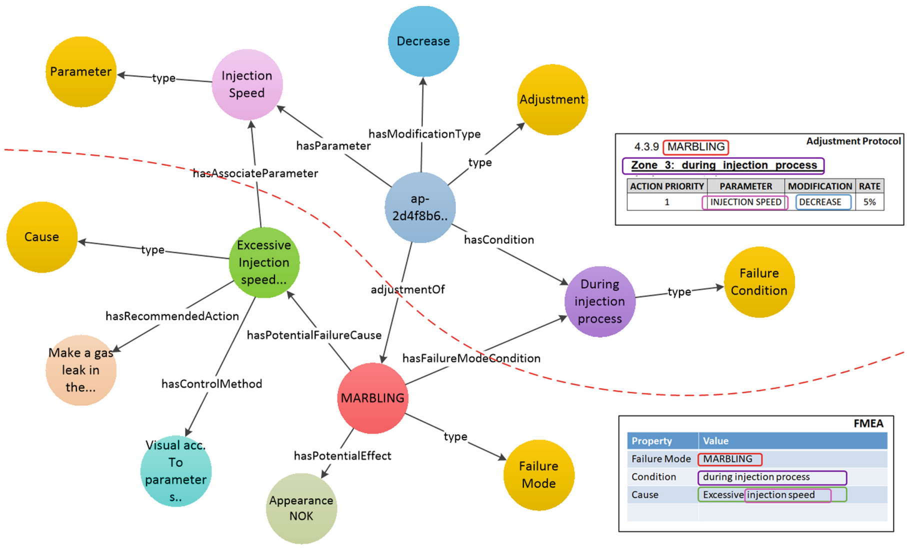

Fig. 4. Knowledge-Graph Output (excerpt) generated from FMEA and IPAP documents.

Table 1. Knowledge graph statistics[a]

	Type	FMEA	IPAP
Axioms	Static	38	73
Class count	Static	4	6
Object property count	Static	4	7
Data property count	Static	3	6
Individual (instance) count	Dynamic	112	222

[a]As per March 2, 2021.

4 Use Case: Shop Floor Parameter Adjustment Recommendation

In this section, we present our shop floor parameter adjustment application enabled by the transformation of heterogeneous semi-structured and non-structured sources to a homogeneous knowledge graph. We also build the foundation to address further issues we raised in Sect. 2. In particular, we migrate from a document-centric to a KG-centric paradigm with explicit semantic relations between the sources, which enables the company to shift tasks in the process towards computer-aided decisions. Figure 5 illustrates the redesigned process, in which the manual tasks from the original process (cf. Fig. 2), which were not computer-supported, have been replaced with computer-supported user tasks. Next, we provide a general description of the use case, followed by an example.

The parameter adjustment procedure makes heavy use of parameter adjustment recommendations, which are retrieved as follows. First, the failure mode

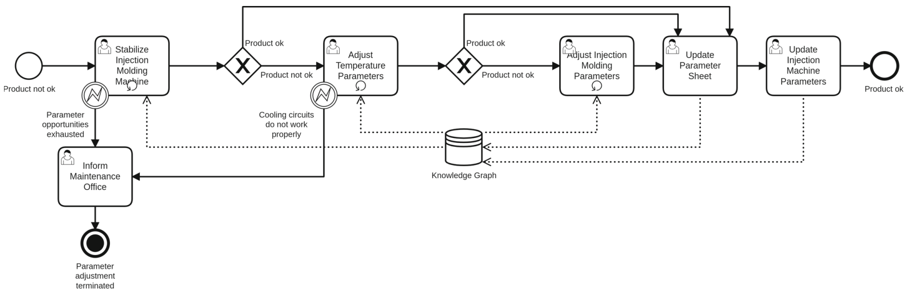

Fig. 5. Redesigned KG-based parameter adjustment process from Fig. 2 - Note that the process consists of user tasks, increasing the degree of automation.

Table 2. Query results for failure mode MARBLING under condition CLOSE TO INJECTION POINT - see Listing 1 for the query.

Failure mode and effects analysis	Adjustment protocol				
Failure cause	Action priority	Modification type	Parameter	Rate	Unit
Excess decompression	1	Decrease	Decompression	1.0	Millimeter
Lack of plasticizing back pressure	2	Increase	Back pressure	1.0	Bar
Excess temperature in hot chamber ...	3	Decrease	Mass temperature	10.0	Celsius

is determined based on automated or manual visual inspection on the shop floor. Next, we obtain a list of prioritized adjustments for the failure using the observed condition. Finally, we retrieve the failure mode, action priority, modification type, parameter, parameter rate, and the unit of the parameter – and sort the results ascending by action priority. The machine operator can then use this information to execute the parameter adjustment.

We illustrate the procedure for the shop floor parameter adjustment recommendation with an example. A machine operator observes a piece with a *Marbling* defect (failure mode) close to the injection point (condition). The operator enters this information on a Human–machine interface (HMI) in order to parameterize the query in Listing 1. The system assists the operator with a list of parameter adjustment recommendations ordered by their action priority - which is the order in which the parameters need to be changed (Table 2). Based on the query result, the operator learns, for instance, that the recommended first adjustment is to decrease the decompression parameter by 1 mm, and the last recommended adjustment is to decrease the mass temperature by 10 Celsius.

Table 2 lists the results for the query in Listing 1, i.e., all *fmea:FailureMode*s and their respective failure conditions and failure causes. The failure cause is crucial, as it connects the FMEA with the IPAP via the parameter that is associated with a failure cause. This parameter is in turn connected to an adjustment, which has a modification type (indicating whether to increase or decrease the parameter), and an adjustment rate. The parameter is also connected to a unit. Finally, the query filters for the observed failure and condition.

```
PREFIX dcterm:<http://purl.org/dc/terms/>
PREFIX ap:
↪  <http://www.w3id.org/teamingai/resources/ont/adjustmentProtocol#>
PREFIX fmea: <http://www.w3id.org/teamingai/resources/ont/FMEA#>

SELECT DISTINCT ?failureCause ?actionPriority ?modifType ?param ?rate
↪  ?unit
WHERE {  ?failure  a fmea:FailureMode;
            fmea:hasFailureCondition ?con;
            fmea:hasPotentialFailureCause ?failureCause.
        ?failureCause fmea:hasAssociateParameter ?param.
        ?adj ap:hasParameter ?param;
            ap:hasModificationType ?modifType;
            ap:actionPriority ?actionPriority;
            ap:adjustmentRate ?rate.
        ?param a ap:Parameter; ap:unit ?unit.
        FILTER regex(str(?failure),"MARBLING")
        FILTER regex(str(?con),"close.+to.+injection.+point")
        ... }
    ORDER BY ASC(?actionPriority)
    LIMIT 3
```

Listing 1. The parameter suggestion query for failure *marbling* under the condition *close to injection point* - see Table 2 for the query results.

5 Discussion

In this section, we discuss the wider context of the parameter recommendation system introduced in this paper. We start by discussing how the knowledge graph and transformation components are part of a larger envisioned software system that aims to translate human teamwork into the digital age. Next, we present our evaluation strategy. Finally, we finish the section discussing current limitations.

The Teaming.AI platform and KGs. The FMEA and IPAP are production system resources for quality management that we lift from a semi-structured to a semantically explicit structured form. This facilitates an increased degree of automation of the parameter adjustment process, and hence also of the production process. With this increase in automation, however, new challenges arise in all software development phases. It is, for instance, unclear how the system should act if it encounters an uncertain situation, and how such situations should be modelled in the first place.

The Teaming.AI platform – of which the artifacts of this paper are part of – addresses these challenges systematically. It is a software system built with human-computer interaction as the guiding principle in all development phases. This is driven by the growing number of tasks software systems can either take over partially or fully from humans - with the main objective to increase productivity and effectiveness. In this context, it is unclear whom of the two should be the performer, and whom the supporter of a task. To address this problem, the human-computer guiding principle is structured with the big five of teamwork [22] and the 4S interdependence framework [11]. These two frameworks are used to analyze and model tasks that require a form of interaction between humans and the software system, in other words who should be the performer and who the supporter. In these tasks, it is important for the human to have confidence in the decisions made by the software system. Knowledge modeling is a major pillar for the Teaming.AI platform, as it needs to support these requirements, and also application requirements. Because of the crucial role KGs have in this software system, it has two components. First, the dynamic KG is responsible for storing high level events, which are aggregated run time events needed for the process-aware IS. And second, the background KG, which is responsible for storing information on *(i)* organizational roles and responsibilities, *(ii)* products, *(iii)* production system resources, and *(iv)* production processes. In this paper, we describe an important building block for the *production system resources* in the background KG.

Evaluation. Overall, we found that the knowledge graph approach offers flexibility and reduces the cost of integrating additional data sources as well as interoperability within the organizations (and potentially beyond) enabled by Semantic Web standards. An evaluation of our approach beyond a qualitative validation through domain experts is out of the scope of the present paper, but we discuss potential quantitative evaluation approaches we plan in future work. To evaluate the system against the currently used approach, we will compare product quality metrics with and without our approach in place. Although quality can be measured in a variety of ways, we can define it pragmatically for our purposes as a fraction of parts meeting the quality standards relative to the total number of produced parts [15]

$$\text{quality} = \frac{\text{saleable parts}}{\text{produced parts}} \tag{1}$$

Another important variable to observe is the time a machine is in production mode, and the average time it takes to resolve an issue. We hypothesize that quality increases as a result of an increase in a productive state, which itself is a result of an increased speed in resolving issues.

Limitations. In this paper, we use a KG as a backbone for a quality management use case within the automotive industry. We validated the two ontologies and the software qualitatively in the context of the current use case of our industrial partner, but leave an investigation into the generalizability to other organizations and use cases for future work. We found that KGs are a useful approach in this context and support a broader vision of KG-enabled teaming, yet they are not the only candidate technology to address the use case requirements. For instance, a relational, key-value, or key-document data base could be used instead of a KG.

They are less flexible and less suitable for the broader vision of enabling Human-AI teaming, but the cost of integration may be similar or lower compared to using a KG. Furthermore, the pool of developers familiar with such technologies is also still larger, although this is not an inherent limitation of the proposed approach. Finally, an evaluation of our approach with the above mentioned evaluation strategies will only provide insights on whether computer aided support had an effect, but not that this effect is specific to the use of a KG.

6 Related Work

This paper contributes to the literature on Semantic Web (SW) technologies in manufacturing, specifically in the context of a real-world quality management use case from the automotive industry. In the following, we review related work on FMEA ontologies.

FMEA Ontologies. One of the first contributions focusing on structured FMEAs are Ref. [6,14,16,21]. Lee et al. [16] introduces the DAEDALUS knowledge engineering framework that integrates product *"design and diagnosis"* with the purpose of *"exchanging and integrating design FMEA and diagnosis models"*. They therefore focus on connecting product design and diagnosis. Along similar lines, [6] discusses two management problems in relation to FMEA, and in knowledge management more generally. They find that *"relevant knowledge may often not be found in an explicit form like databases"*, and that *"the access to knowledge is encumbered with the problem that different actors use different terms to talk about the same topic"*. The authors conclude that *"It [ontologies] can solve the main shortcomings and the resulting problems as mentioned"*. Ref. [14] focus on the second problem raised in [6] and use ontologies to map different functional models to FMEA sheets. A number of approaches also have been developed that aim to make unstructured FMEAs information more structured and semantically explicit. For instance, [26] introduce an FMEA knowledge graph with ontologies in manufacturing processes; [17] describes an FMEA process and software tools for a lead-free soldering process; finally, [25] uses ontologies to ease sharing, reuse, and maintenance of FMEAs in manufacturing processes.

Similar to the approach in this paper, [20] address the problem of using natural language text for FMEAs, which makes it difficult to reuse this knowledge. We refer in this paper to documents in natural language text as non-structured documents. The organization as a result wasted resources on the FMEA document. Our work complements this and is also motivated by waste. In addition, we show how a structured and semantically explicit representation can help to increase the degree of automation. Another similarity to our work is the proposal of an FMEA ontology. Ref. [7] also state that the problem of FMEAs is *"in the form of textual natural language descriptions that limit computer-based extraction of knowledge for the reuse of the FMEA analyses in other designs or during plant operation."*. They also stress the need to move from non-structured to structured FMEAs and base their ontology on ISO-15926 to define general terms as a foundation so *"engineers can build new concepts from the basic set of concepts"*. Finally, Ref. [8] point out the importance of connecting a systems functional dependency description with the FMEA – using the heating, ventilation and cooling system example from the ISO 60812:2006 standard. Including functional dependencies enables *"automated reasoning to test and infer dependencies"* – which we have not considered so far, but will cover in future work.

7 Conclusion and Future Work

The main contribution of this paper is the integration of Injection Process Adjustment Protocol (IPAP) and Failure Mode and Effects Analysis (FMEA) knowledge; and the parameter recommendation applications this enables. We motivated the importance of this problem economically. First, the objective is to reduce waste caused by incorrect parameters. Second, the training of injection technicians is time-intensive and therefore costly and may take up to one year. Third, the long term objective is to fully automate the parameter adjustment process. We show how the integrated knowledge can be used by a parameter adjustment recommendation engine within a process-aware IS, and leave the application for conformance checking as future work. In particular, we show how the parameter adjustment recommendation engine changes the injection parameter adjustment process by switching from manual tasks to user tasks. This, combined with the conformance checking and other systems, may increase the automation even further, for example towards a fully automated process where humans have the supporter role and the software system the performer role.

To this end, we introduced a parameter adjustment protocol ontology and integrated it with a FMEA ontology - which we adapted to our purposes and is introduced in Ref. [20]. Both ontologies are based on a set of documents we received from our industrial partner. We accompany these ontologies with a software library for transforming spreadsheets. We argue that this is important, as it lowers the entry costs to these ontologies for third parties. These artifacts are the foundation for a KG that will be used as the backbone for AI applications for quality management and beyond. The KG can, for example, integrate sensor data from the injection moulding machine, and can itself be an input for machine learning models.

For future work, we plan to integrate data from the injection moulding machine via the parameter class. Specifically, we plan to integrate two sources: *(i)* the parameter settings from the machine, and *(ii)* the actual parameter values observed by sensors within the machine. Integrating the disparate, heterogeneous, and largely unstructured collection of information sources relating to FMEA and IPAP enables applications beyond the parameter adjustment recommendation engine we present here. Conformance checking, for instance, can leverage the KG to compare actual versus recommended parameter changes, which may reveal an incomplete and partially incorrect adjustment protocol. Furthermore, we used BPMN so far as a process modeling language to document and redesign the parameter adjustment processes, but not as part of the developed system. In future work, we aim to add process context to the KG, for which we aim to adopt the concept of a modular KG to organize different contexts [1]. This is linked to the idea of a layered KG, where a KG has different meanings to different stakeholders - which we plan to explore [3]. Finally, adding the just mentioned process context and using it as an active system component is an important enabler towards higher degrees of automation using service or script tasks. Using the process model actively is also important as it makes the performer and supporter roles mentioned in the previous paragraph usable in production. Maybe even more important, it makes these roles explicit for all stakeholders.

Acknowledgements. This work received funding from the Teaming.AI project in the European Union's Horizon 2020 research and innovation program under grant agreement No. 95740.

References

1. Bachhofner, S., Kiesling, E., Kabul, K., Sallinger, E., Waibel, P.: Knowledge graph modularization for cyber-physical production systems. In: International Semantic Web Conference (Poster), Virtual Conference, October 2021
2. Bellomarini, L., Fakhoury, D., Gottlob, G., Sallinger, E.: Knowledge graphs and enterprise AI: the promise of an enabling technology. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macau SAR, China, pp. 26–37. IEEE, April 2019
3. Bellomarini, L., Sallinger, E., Vahdati, S.: Knowledge graphs: the layered perspective. In: Janev, V., Graux, D., Jabeen, H., Sallinger, E. (eds.) Knowledge Graphs and Big Data Processing. LNCS, vol. 12072, pp. 20–34. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53199-7_2
4. Bozdana, A., Eyercioglu, Ö.: Development of an expert system for the determination of injection moulding parameters of thermoplastic materials: EX-PIMM. J. Mater. Process. Technol. **128**(1), 113–122 (2002)
5. Buchgeher, G., Gabauer, D., Martinez-Gil, J., Ehrlinger, L.: Knowledge graphs in manufacturing and production: a systematic literature review. IEEE Access **9**, 55537–55554 (2021)
6. Dittmann, L., Rademacher, T., Zelewski, S.: Performing FMEA using ontologies. In: 18th International Workshop on Qualitative Reasoning, Evanston, IL, USA, pp. 209–216, August 2004
7. Ebrahimipour, V., Rezaie, K., Shokravi, S.: An ontology approach to support FMEA studies. Expert Syst. Appl. **37**(1), 671–677 (2010)
8. Hodkiewicz, M., Klüwer, J.W., Woods, C., Smoker, T., Low, E.: An ontology for reasoning over engineering textual data stored in FMEA spreadsheet tables. Comput. Ind. **131**, 103496 (2021)
9. Hubauer, T., Lamparter, S., Haase, P., Herzig, D.M.: Use cases of the industrial knowledge graph at siemens. In: International Semantic Web Conference (P&D/Industry/BlueSky), Monterey, CA, USA, October 2018
10. ISO Central Secretary: Quality management systems - particular requirements for the application of ISO 9001:2000 for automotive production and relevant service part organizations. Standard ISO/TS 16949:2009 (2009)

11. Johnson, M., Vera, A.: No AI is an island: the case for teaming intelligence. AI Mag. **40**(1), 16–28 (2019)
12. Kagermann, H., Wahlster, W., Helbig, J., et al.: Recommendations for implementing the strategic initiative Industrie 4.0: final report of the Industrie 4.0 working group. Technical report, Berlin, Germany (2013)
13. Klingenberg, C.O., Borges, M.A.V., Antunes, J.A.V., Jr.: Industry 4.0 as a data-driven paradigm: a systematic literature review on technologies. J. Manuf. Technol. Manag. (2019)
14. Koji, Y., Kitamura, Y., Mizoguchi, R., et al.: Ontology-based transformation from an extended functional model to FMEA. In: 15th International Conference on Engineering Design, pp. 323–324. Melbourne, Australia, August 2005
15. Kronos Incorporated: Overall Labor Effectiveness (OLE): Achieving a highly effective workforce (2007). https://workforceinstitute.org/wp-content/uploads/2008/01/ole-achieving-highly-effective-workforce.pdf. Accessed: 02 June 2022
16. Lee, B.H.: Using FMEA models and ontologies to build diagnostic models. AI EDAM **15**(4), 281–293 (2001)
17. Molhanec, M., Zhuravskaya, O., Povolotskaya, E., Tarba, L.: The ontology based FMEA of lead free soldering process. In: 34th International Spring Seminar on Electronics Technology (ISSE), Tratanska, Lomnica, Slovakia, pp. 267–273. IEEE, May 2011
18. Noy, N.F., McGuinness, D.L., et al.: Ontology development 101: a guide to creating your first ontology (2001)
19. OMG: Business Process Model and Notation (BPMN), Version 2.0.2, December 2013. http://www.omg.org/spec/BPMN/2.0.2
20. Rehman, Z., Kifor, C.V.: An ontology to support semantic management of FMEA knowledge. Int. J. Comput. Commun. Control **11**(4), 507–521 (2016)
21. Russomanno, D.J., Bonnell, R.D., Bowles, J.B.: Functional reasoning in a failure modes and effects analysis (FMEA) expert system. In: Annual Reliability and Maintainability Symposium 1993 Proceedings, Atlanta, GA, USA, pp. 339–347. IEEE, January 1993
22. Salas, E., Sims, D.E., Burke, C.S.: Is there a "big five" in teamwork? Small Group Res. **36**(5), 555–599 (2005)
23. Stamatis, D.H.: Failure Mode and Effect Analysis: FMEA from Theory to Execution. Quality Press, Milwaukee (2003)
24. Wu, Z., Liu, W., Nie, W.: Literature review and prospect of the development and application of FMEA in manufacturing industry. Int. J. Adv. Manuf. Technol. **112**(5–6), 1409–1436 (2021)
25. Xiuxu, Z., Yuming, Z.: Application research of ontology-enabled process FMEA knowledge management method. Int. J. Intell. Syst. Appl. 34–40 (2012)
26. Zhao, X., Zhu, Y.: Research of FMEA knowledge sharing method based on ontology and the application in manufacturing process. In: 2nd International Workshop on Database Technology and Applications, Wuhan, China, pp. 1–4. IEEE, November 2010
27. Zhou, B., Svetashova, Y., Pychynski, T., Kharlamov, E.: Semantic ML for manufacturing monitoring at Bosch. In: International Semantic Web Conference (Demos/Industry), Virtual Conference, November 2020