








# Automated Process Knowledge Graph Construction from BPMN Models

Stefan Bachhofner<sup>1</sup>(✉) , Elmar Kiesling<sup>1,2</sup> , Kate Revoredo<sup>1</sup> ,  
Philipp Waibel<sup>1</sup> , and Axel Polleres<sup>1,3</sup> 

<sup>1</sup> Institute for Data, Process and Knowledge Management,  
Vienna University of Economics and Business, Vienna, Austria  
{Stefan.Bachhofner,Elmar.Kiesling,Kate.Revoredo,Philipp.Waibel,  
Axel.Polleres}@wu.ac.at

<sup>2</sup> Austrian Center for Digital Production, Vienna, Austria

<sup>3</sup> Complexity Science Hub Vienna, Vienna, Austria

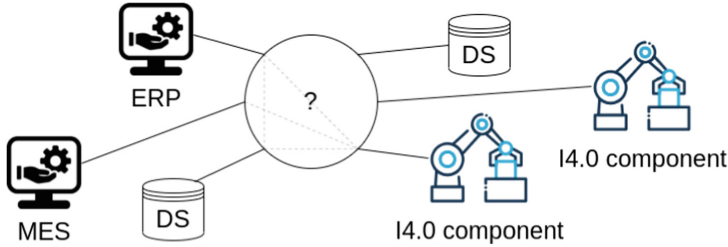
**Abstract.** Enterprise knowledge graphs are increasingly adopted in industrial settings to integrate heterogeneous systems and data landscapes. Manufacturing systems can benefit from knowledge graphs as they contribute towards implementing visions of interconnected, decentralized and flexible smart manufacturing systems. Process knowledge is a key perspective which has so far attracted limited attention in this context, despite its usefulness for capturing the context in which data are generated. Such knowledge is commonly expressed in diagrammatic languages and the resulting models can not readily be used in knowledge graph construction. We propose BPMN2KG to address this problem. BPMN2KG is a transformation tool from BPMN2.0 process models into knowledge graphs. Thereby BPMN2KG creates a frame for process-centric data integration and analysis with this transformation. We motivate and evaluate our transformation tool with a real-world industrial use case focused on quality management in plastic injection molding for the automotive sector. We use BPMN2KG for process-centric integration of dispersed production systems data that results in an integrated knowledge graph that can be queried using SPARQL, a standardized graph-pattern based query language. By means of several example queries, we illustrate how this knowledge graph benefits data contextualization and integrated analysis. In a broader context, we contribute towards the vision of a process-centric enterprise Knowledge Graph (KG). BPMN2KG is available at <https://short.wu.ac.at/BPMN2KG>, and the sample queries and results at <https://short.wu.ac.at/DEXA2022>.

**Keywords:** Business process management · BPMN 2.0 · Semantic web · Knowledge graph · Model transformation · 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.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022  
C. Strauss et al. (Eds.): DEXA 2022, LNCS 13426, pp. 32–47, 2022.  
[https://doi.org/10.1007/978-3-031-12423-5\\_3](https://doi.org/10.1007/978-3-031-12423-5_3)



**Fig. 1.** Motivation: Dispersed data and unknown dependencies across the automation hierarchy (ERP, MES), data stores (DS), and production system components (I4.0 component).

## 1 Introduction

Relating domain and process knowledge to disparate and heterogeneous data is a challenge in most enterprise settings, which is particularly pronounced in the data-rich context of Cyber-physical Production Systems (CPPSs). Such systems currently drive a paradigm shift in manufacturing that is alluded to as the fourth industrial revolution and associated with the term Industry 4.0 (I4.0) [20]. This fundamental shift in industry is inherently driven by data [22] and characterized by requirements for flexible, networked, and self-configurable processes [32]. Consequently, data and process landscapes are expanding rapidly in smart manufacturing, but they typically remain disparate and fragmented (i) across information systems and data stores [15], (ii) between office and shop floor environments, and (iii) across business functions. Figure 1 illustrates this disconnect between various information systems across the automation hierarchy, data stores, and production system components. This disconnect raises the following challenges,

- C1** *Integration* across multiple organizational, functional, and temporal levels of granularity,
- C2** *Contextualization* of raw sensor data with higher-level operational information and quality requirements, and
- C3** *Linking and aggregation* of decisions and goals on the production and operational levels to higher-level business goals.

KGs – which are characterized by a flexible schema, decentralized architecture, and ability to support data and knowledge integration – provide a promising foundation for such challenges. To integrate the fragmented process and data landscape through KGs, however, it is necessary to consider the process context. To address these challenges we therefore propose a combination of (i) Business Process Modeling, which was proposed as a method to tackle fragmentation challenges in manufacturing [1] and (ii) KG modeling based on Semantic Web (SW) standards, which have recently shown promising results in I4.0 applications [7, 26, 29]. We specifically propose BPMN2KG<sup>1</sup> as a tool

<sup>1</sup> BPMN2KG is available at <https://short.wu.ac.at/BPMN2KG>.

to automatically transform business process models in Business Process Model and Notation (BPMN) 2.0 [25] into a KG representation in RDF, based on and extending an existing ontology for process representation [5]. The transformation tool currently supports 33 BPMN 2.0 elements, including the most used ones according to [33].

Our motivation stems from multiple industrial applications within the H2020 Teaming.ai project<sup>2</sup>. The real-world scenario we selected to illustrate and validate our approach in this paper focuses on quality management and optimization of injection molding processes in the automotive industry. This use case illustrates how the proposed approach combines heterogeneous manufacturing data and process landscapes by integrating domain-specific and semantic abstraction models. BPMN2KG contributes towards using untapped process knowledge for integration initiatives using process graph modeling. It facilitates integrated querying of manufacturing data and process knowledge with SW methods and tools. This creates the ability to easily link data across sources and with manufacturing domain knowledge and provides a foundation for process-centric enterprise KG construction in I4.0 and beyond.

The remainder of the paper is structured as follows: Sect. 2 provides an introduction to process knowledge representation and knowledge graphs; Sect. 3 introduces the problem by means of a real-world quality management focused use case in an industrial setting; Sect. 4 introduces BPMN2KG and covers requirements, architecture, and implementation details; and Sect. 5 shows the results of the questions raised by our quality management focused use-case. Section 6 details related work. Finally, the paper concludes with remarks on BPMN2KG for our use case in Sect. 7.

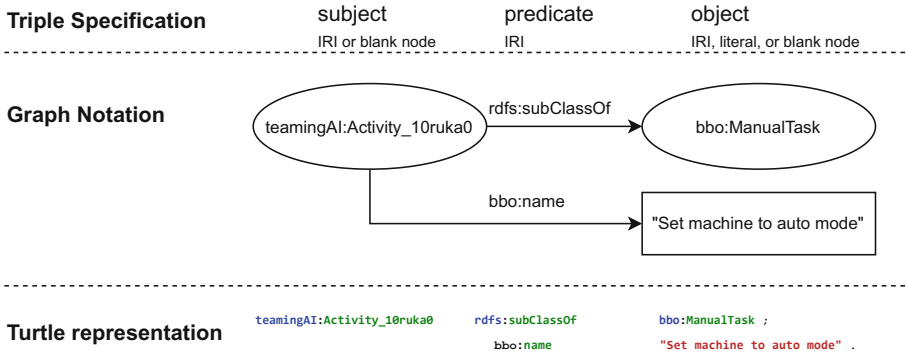
## 2 Background

*Process knowledge representation in Industry 4.0.* A business process is a sequence of events, activities, and decision points that involve a number of actors and objects and leads to an outcome that is of value to at least one customer. It is typically represented in graphical models [19]. In recent years, BPMN 2.0 [25] has become a de-facto standard for modeling business processes, and it has also attracted increasing attention in the manufacturing domain [2, 4]. It provides a wide range of graphical syntax elements that allow to describe process aspects in semantically well-defined terms, in various complexities, and for different use cases. eXtensible Markup Language (XML) is commonly used as a data format for BPMN. Due to space constraints, we do not discuss individual BPMN elements here; the full specification can be found in [25].

*Knowledge Graphs.* A KG is "a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities." [18]. Initially developed in the

---

<sup>2</sup> <http://teamingai-project.eu>.



**Fig. 2.** RDF triple [13] (top) example in graph notation (middle) and turtle representation (bottom).

context of the Semantic Web (SW), KGs have today seen widespread adoption in web technology companies such as Microsoft, Google, Facebook, IBM [24], and Apple [23], where they provide an infrastructure to support services such as search, recommendations, and automation. KGs rely on graph data models such as labeled property graphs or directed edge-labelled graphs [18].

For modeling KGs, Resource Description Framework (RDF) [13] is a widely used language recommended by the World Wide Web Consortium (W3C). KGs in RDF are formed from triples, each of which consists of a subject, a predicate, and an object. Figure 2 illustrates two triples in RDFs from our motivating use case - the two triples belong to the first task of our mass production process. The first triple encodes the statement “*Activity 10ruka0* is a *subclass* of *bbo:ManualTask*”, and the second “*Activity 10ruka0* has the *bbo name* of *Set machine to auto mode*”. In graph notation, IRIs and blank nodes are represented with an ellipse and literals with a rectangle.

*BPMN-based Ontology (BBO)* [5] is an ontology to represent business processes modeled in BPMN 2.0 in a KG. An ontology is necessary as it for example allows us to sub-class its concepts, or use their properties. We see this in the example above from Fig. 2, where we define the thing that is identified by the Universal Resource Identifier *teamingAI:Activity\_10ruka0* as a sub-class of *bbo:ManualTask*, and use the property *bbo:name* to give the Universal Resource Identifier (URI) a name. In addition to standard BPMN elements, BBO also provides some non-standard elements, such as a description in which manufacturing facility the process should be executed. We will use BBO as a basis and extend it with additional BPMN elements that are not covered in BBO.

### 3 Industrial Use Case

Our research into process knowledge graphs is motivated by three real-world industrial use cases in the context of the H2020 project Teaming.ai, which aims to develop a Human-AI Teaming Platform for Maintaining and Evolving AI Systems in Manufacturing. The machine-interpretable representation of process knowledge is crucial both for data integration and the human-centered collaboration of human and AI agents in I4.0 scenarios.

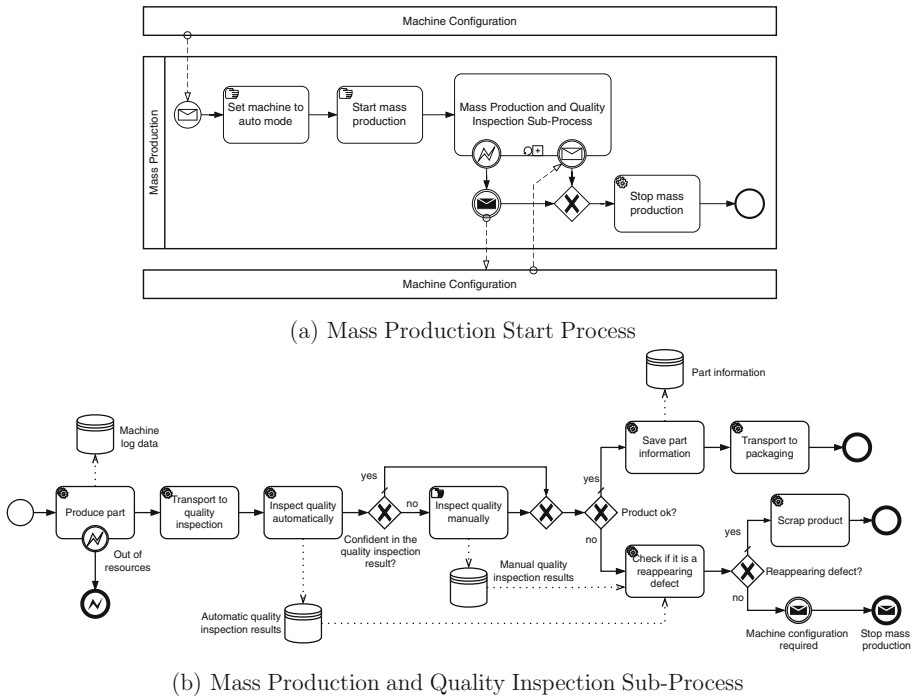
In this paper, we focus on a use case provided by a major supplier in the automotive industry specializing in plastic injection moulding. The production of plastic parts requires multiple processes, each defined in a separate process model: (i) First, the production material – the plastic granules – is prepared, which involves inspecting its quality. If quality is approved, the granules are fed to the manufacturing machine, otherwise, the material supplier is informed. (ii) Next, the manufacturing machine is configured by setting various machine parameters. These settings are then tested by producing a trial part and inspecting its quality. If the quality of the part is not satisfactory, the machine parameters are further readjusted, and another trial part is produced. This is continued until the quality meets the requirements. (iii) Finally, mass production starts with the determined machine parameter settings.

These processes are linked directly and indirectly through shared objects and data flows. They are also linked to other processes not considered in our use case scenario, such as mold engineering and setup, logistics processes, inventory handling, or order handling.

Figure 3 depicts the mass production process in BPMN 2.0. Figure 3a defines the start of the mass production and Fig. 3b illustrates the production process itself, together with the quality inspection of the produced part, as a sub-process. This sub-process is repeated for each unit produced until a stopping event is received or an error occurs.

The mass production process starts with the production of a part using an injection molding machine. During this step, a wealth of machine log data are generated and stored in a database. In our use case scenario, this log data will be used to populate a KG. Next, the quality of the part is checked by means of an automatic Visual Quality Inspections (VQIs) system. If this VQI system is not confident about its result (determined by a confidence threshold), a human-based manual inspection takes place. The result of both checks are again stored in a database. The quality of the part then determines the next activity. If the quality is *ok*, additional information about the part, e.g., part id, is persisted, and the part is handed over to packaging. If the quality is *not ok*, the next step depends on whether or not it is a recurring defect. In case of a recurring defect, mass production stops and a reconfiguration of the machine parameters is requested. For non-recurring defects, the part is scrapped and the next part is produced.

To produce in high volume, the company uses several manufacturing machines. Not all of these machines are of the same type, produce the same quality, and do not have the same capabilities or parameters. Moreover, some



**Fig. 3.** Mass production process of plastic parts industrial use case in BPMN 2.0.

production parts require unique treatments, for example a special finishing, and for some parts, automated VQI is not feasible.

Thus, a wide range of specialized variants of the discussed processes are used by the company, resulting in an extensive process landscape with many different process models. In addition, the execution of these processes creates vast amounts of data, including parameter and sensor data from the injection machines, energy and water sensor data, quality inspection data, and part information for each part produced. These persisted data are used for different purposes, for example in the design of new production process models and products/molds, to optimize the production process by analyzing the quality inspection results, or when performing the machine parameter setup.

A key challenge in this context is the fragmented nature of the data produced, which are not contextualized or linked to process knowledge. This makes it difficult to answer common questions such as:

- (Q1) Across process models, which activities store data in or consume data from the various data stores?
- (Q2) Which production processes include a quality control activity (of any kind)?
- (Q3) What are the observed defect rates per defect type, across all production variants, for manual versus automatic visual quality inspection?

**(Q4)** What are the machine log data for produced units that exhibit a particular type of defect in manual or automated quality inspection?

Some of these questions require only information from a single data source, whereas others require combined knowledge derived from data stored in multiple systems and process models. The integrated querying capabilities across process models, data sources, and domain knowledge enabled by the KG-based approach is particularly beneficial in these latter cases.

## 4 BPMN2KG Tool

In the following, we summarize the requirements we elicited from our industrial use cases (Sect. 4.1), outline the KG construction with BPMN2KG on a schematic level (Sect. 4.2), and finally discuss implementation aspects (Sect. 4.3).

### 4.1 Requirements

Informed by the use case introduced in Sect. 3 as well as other use cases in the manufacturing domain as part of the Teaming.ai project, we collected the following set of requirements for business process model and KG integration in several rounds of workshops with domain experts:

**(R1) Flexible semantic data model and schema:** Supporting integration of process knowledge with domain knowledge and data requires a flexible model that can express relations between resource, process, and data elements. The tool shall not extend existing process modeling tools to encode explicit semantics into BPMN models, but rather impose semantics on the schema level through KG construction, curation, and completion techniques.

**(R2) Automated model transformation:** BPMN2KG shall automatically transform any valid BPMN model into a KG representation. All core as well as the most widely used other BPMN elements shall be supported. We will base the choice of these elements on studies such as [33], which found that only 20% of BPMN syntax elements are regularly used in their sample. In particular, the syntax elements to be supported are, in descending order of popularity, task, sequence flow, start event, end event, gateway, parallel gateway, data-based eXclusive OR (XOR) gateway, pool, and lane.

**(R3) Rich process-oriented querying** across functional areas, heterogeneous data sources, model and instance data and the process hierarchy. This necessitates both navigational queries, e.g., to express precedence patterns along the sequence flows, and graph-pattern based queries, e.g., to search for specific matches such as the use of particular data across administrative, support, and production-level processes (cf. the motivating questions Q1-Q4).

**(R4) Modularity and extensibility:** Whereas the prototype shall support basic transformation of the process structure of any valid BPMN 2.0 model, it should be modular and extensible through custom mappings. Due to this extensibility, the tool shall also be universally applicable beyond the manufacturing domain.

## 4.2 Process Knowledge Graph Construction

BPMN2KG constructs a KG from BPMN models, transforming multiple isolated models into a uniform representation that can be queried, linked to background knowledge, and integrated with instance data in a single, integrated graph. We call this KG a *process knowledge graph*, as it contains explicit process knowledge. The tool thereby makes this knowledge accessible for other systems via widely used standards. Figure 4 illustrates this concept by means of our mass production process (top). The red and green rectangles relate the graphical elements to their XML (middle) representation and show how these elements – after the transformation with BPMN2KG – are represented in RDF (bottom). Let us consider the first task of the process (green rectangle), which is the manual task “*Set machine to auto mode*” marked with a hand in the upper left corner. This task is represented in XML with the `bpmn:manualTask` tag, and has the two attributes `id` and `name` with values “*Activity\_1e93nvv*” and “*Set machine to auto mode*”. BPMN2KG transforms these two attributes into the two triples (**(`tai:Activity_1e93nvv` `rdfs:subClassOf` `bbo:ManualTask`)** and (**(`Activity_1e93nvv` `bbo:name` “*Set machine to auto mode*”`@en`)**). We use unit-tests to verify the correctness of such transformations.

To transform BPMN models into a knowledge graph representation, we use RDF Mapping Language (RML) as a declarative mapping language. Declarative software languages enable a higher level of abstraction – consider for example the Structured Query Language (SQL), another declarative language, where we define data structures without worrying about their physical realization, and queries without worrying about their procedural execution. RML allows for a similar abstraction for the relationship of heterogeneous data structures to RDF. RML by definition is a “*a generic mapping language, based on and extending*” the Relational data base to Resource description framework Mapping Language (R2RML) standard [12]. R2RML is a W3C recommendation [14], but is specialized for “*relational databases to RDF datasets*” [14]. To transform BPMN into a KG representation, we use RML to define a relation from XML to RDF for each XML element. RML uses XPath [10] to create logical sources that are mapped to one or more RDF triples. For the mapping to BBO, this results in 23 `rr:TriplesMap` definitions<sup>3</sup>.

*Supported BPMN Elements.* A study that analyzed 120 BPMN diagrams found that only 20% of BPMN syntax elements are regularly used in their sample [33]. These syntax elements are, in descending order of popularity, task, sequence flow, start event, end event, gateway, parallel gateway, data-based XOR gateway, pool, and lane. Based on this observation, we decided that the first version of the tool has to support these syntactic elements. Unfortunately, BBO [5] does not include classes for pools and lanes. For this reason, we extended BBO into Business process model and notation Based Ontology Extension (BBOExt). In addition to pools and lanes, this extension supports message flows, association,

<sup>3</sup> <https://short.wu.ac.at/BPMN2BBO>, <https://short.wu.ac.at/BPMN2BBOExt>, and <https://short.wu.ac.at/BPMN2BBOExtANDBBO>.



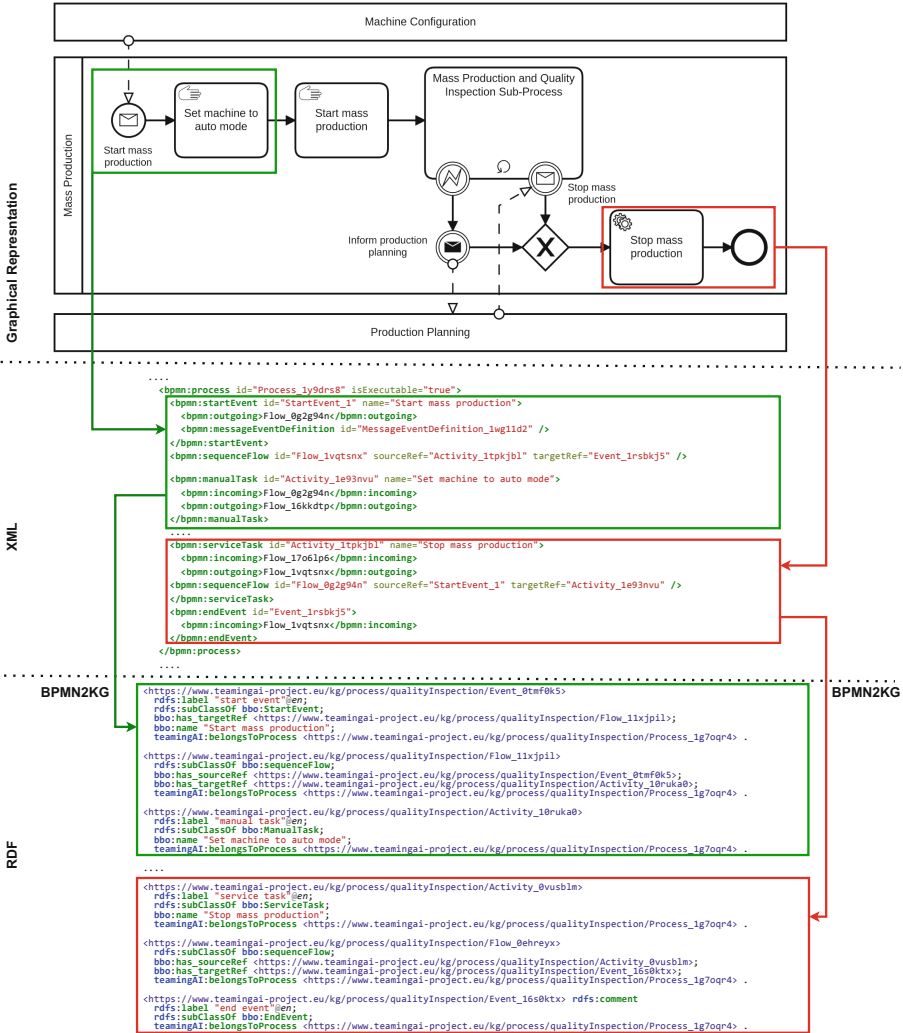


Fig. 4. Mapping example for excerpts of mass production process.

text annotation, data object, data object reference, data output association, data input association, and data store reference.

### 4.3 Implementation

*Command Line Tool.* Business Process Model and Notation to Knowledge Graph (BPMN2KG) (See footnote 1) is implemented as a command line tool in Python<sup>4</sup> with five arguments, two of them are required: `--bpmn-input`

<sup>4</sup> <https://www.python.org/>.

(the BPMN file or a directory) and `--kg-output` (the RDF file that the KG should be saved to or a directory where files should be saved to) that point BPMN2KG to the two files needed for the transformation. The other three arguments let the user choose the ontology (`--ontology`), the subject's URI template (`--uri-template`), and the serialization format of the output file (`--serialization-format`).

*Benefits.* The software layer on top of the RML adds a number of convenient features. First, the user can specify a folder instead of only a single file at a time for the transformation. Second, the URI templates need to be set only once and do not need to be manually exchanged each time. Third, it allows the user to easily change the target ontology. And finally, it encapsulates the complexity of RML into simple command line calls. As an engine for the RML transformations, we decided to use RMLMapper<sup>5</sup> as it offers a command line as well as a library interface which enables us to change to a different architecture in the future without changing the technology behind the transformation. We can for example change from Python to Java without replacing the engine. And finally, BPMN2KG can be integrated with any BPMN2.0 (which we follow) compliant software.

## 5 Application Scenarios

In this section, we focus on the use case introduced in Sect. 3, which tackles quality management and analytic challenges in plastic injection moulding. Specifically, we transform the graphical knowledge on the processes involved in production setup, execution, and quality control – which are captured in several BPMN models – into a KG representation. We enrich the KG with information about the manufacturing machines used by the various process activities and the involved resources and validate the capability to contextualize data in a KG with the transformed process knowledge.

Next, we illustrate how the resulting KG supports sensor data contextualization and analysis in quality management - addressing the previously identified requirements in Sect. 4.1 via the examples queries introduced in Sect. 3. See Appendix A for the queries pertaining ( $Q1$ ) to ( $Q3$ ), their results, and the link to all queries - including ( $Q4$ ) and the full syntax.

*(Q1) Data Flow Analysis.* A common challenge in complex production systems – as well as information systems more generally – is the proliferation of heterogeneous systems and dispersed data stores. The lack of a (at least high-level) overview of data flows makes it difficult to trace data provenance as well as to understand the complex interdependencies that exist between various process activities, systems, and data stores. Modeling data flows in BPMN using

<sup>5</sup> RMLMapper: <https://github.com/RMLio/rmlmapper-java> with commit 54bf875.

*data store* and *data association* elements is helpful to document such relationships, but the resulting process models can not be readily queried and connections across process models are not visible. The integrated KG produced by BPMN2KG can help to untangle data flows and dependencies in large process landscape.

Specifically, example query (Q1) illustrates how the KG can support integrated querying of data flows from and to data stores. The query retrieves all data associations between activities and data stores (cf. Fig. 5) and indicates the direction of the flow. The result shows, for instance, that the activity “*Check if it is a reappearing defect*” consumes data from the data store “*Manual quality checking result*”, while the activity “*Check quality manually*” writes to it. The graph-based structure also provides a foundation for more complex object-centric analyses of data flows across process models and highlights how the KG can contribute towards mapping the process and data landscapes.

(Q2) *Cross-Model Activity Querying*. Process knowledge becomes even more useful once the model elements are associated with semantic concepts. For instance, abstraction hierarchies across activities allow for efficient querying. In our use case, for instance, automated and manual quality inspection activities are all subclasses of *tai:QualityManagementActivity*. Therefore, it is possible to use the domain knowledge captured in the activity model in the queries. (Q2) selects all activities that are sub classes of *tai:QualityManagementActivity* and return their IDs, the name of the activity, and the ids of their respective processes (cf. ??). The result shows four activities related to quality management in three different processes.

(Q3) *Comparing Detection Rates of Manual and Automated Quality Inspection*. Beyond interlinking process models and associating them with domain knowledge, the process KG can also link process models to instance data such as quality inspection results. In our use case, which focuses on quality management, this can be used to investigate observed defects by defect type, for different types of quality inspection activities, and across process variants. The query in ??, for example, aggregates the observed cases for different defect types across processes and groups them by task type and defect type. Note here how we use BPMN2KG to contextualize quality management data with respect to process knowledge.

(Q4) *Retrieving Machine Log Data for Defects*. As a final illustrative application scenario, the process KG references instance-level machine log data and makes it available for process-oriented querying<sup>6</sup>. This makes it possible to contextualize and retrieve sensor readings when diagnosing quality issues. For instance, the SPARQL query and result for (Q4) retrieves the machine log data for produced units with a particular type of defect in manual or automated quality inspection. The query in particular retrieves all defect parts with their product ID, product name, part ID, and stroke measurements for cushion, plasticisation, and transfer.

---

<sup>6</sup> You can find the query (Q4) at <https://short.wu.ac.at/DEXA2022-Q4>.

## 6 Related Work

Various KG applications within business process management have been developed in the literature, including KGs as a means to support process modeling [11], process model querying [27], and event log generation [8]. In this paper, we present a transformation tool for process models into a KG representation, with a focus on the integration with raw data and domain knowledge.

Reference [9] proposes a modeling language that combines process and domain knowledge. The approach set up a hybrid knowledge base derived from diagrammatic models, semantically lifted legacy data and open geospatial data. The specification is done manually. The proposed vision is similar to BPMN2KG, as it also aims to create an integrated semantic data fabric for process model contents and contextual data. However, the focus is not on any particular business process modeling language and the authors do not provide a mechanism to transform these models into a KG representation. They instead propose to integrate the semantic model into a customised BPMN front end.

Motivated by cost reduction through the reuse of data from legacy systems, [21] propose an approach for the transformation of BPMN models into OWL2 ontologies. Similar to [5], this work does not provide automatic integration of process and domain knowledge in a single representation.

In a similar domain as the one tackled in this paper, [30] models industrial business processes for querying and retrieval using OWL and SWRL. This also results in a semantic representation of business process models; key differences are the more limited set of transformed BPMN elements (they do not include data sources) and the use of OWL as a representation formalism. Furthermore, the paper does not address the integration of production systems data.

Reference [28] aimed to semantically annotate process models at design time. This is accomplished in a four step process. Similarly to our work, they also map, for example, an activity in a process model to an entity in a KG. However, this approach is based on Event-driven Process Chains (EPCs) [31] rather than BPMN. Another major difference is the execution of the four-step process at *design-time*. Our approach does not focus on assistance during modeling, but transforms models for integration and contextualization at *execution time*. Hence, their work is complementary to ours.

Similar to our work, [16] construct a KG for CPPS. They focus on KG construction from multiple design perspectives to achieve integration among these. However, this introduces uncertainty as different design perspectives might model the same construct differently, or leave it out completely. This is different to our work since we have no uncertainty as we have one perspective, the process perspective. Their work is hence complementary to ours.

## 7 Conclusions

In this paper, we introduce BPMN2KG to integrate process knowledge, domain knowledge, and dispersed data into a KG representation. We motivate the need

for the approach by challenges that arise in the context of an I4.0 use case – which requires flexible processes, has a large process variety, and has to cope with increased “datafication” of the shop floor. BPMN2KG eases (i) the integration across multiple views and granularity levels using data stores (**C1**), (ii) the contextualization by adding process context to data (**C2**), and (iii) the linking and aggregation of production and operational levels (**C3**) - which we illustrate by the example of a plastic injection molding and quality management use case.

Our automatic transformation can further replace a manual semantic annotation of process models, which is generally not feasible in the face of large process landscapes [11]. Additionally, we provide the means to answer complex questions that require the combined knowledge of lower-level shop floor data and higher-level process information. Our work also contributes towards the vision of a process-centric, or at least a process knowledge informed enterprise KG. This is linked to the concept of layered KGs for CPPS presented in [6], where a KG has different domain views (for example process engineering and quality control) and layers based on Reference Architectural Model Industrie 4.0 (RAMI 4.0), which are decoupled I4.0 layers. And finally, as a minor contribution, we open source the RML rules which map BPMN models to the ontologies in our public repository, which means they can be used freely by anyone.

In future work, we plan to build upon and extend the current transformation tool. First, in the present paper we assume that the process logs and the machine logs are available in triple format. For the former, we are indeed already working on an accompanying transformation tool for XES<sup>7</sup>. This software tool will be used alongside BPMN2KG in a software system called Teaming.AI [17]. Beyond, we evaluate the usefulness of other target process ontologies, such as BPMN [3], and an extension that allows for a transformation from the KG to a BPMN XML model.

**Acknowledgement.** This work has also received funding from the Teaming.AI project in the European Union’s Horizon 2020 research and innovation program under grant agreement No 95740.

## A SPARQL Queries and Results

Due to space constraints, we deleted all prefix statements in the following queries and completely exclude (*Q2*), (*Q3*), and (*Q4*) – you can find all queries with the full syntax and the results at <https://short.wu.ac.at/DEXA2022>.

---

<sup>7</sup> <https://git.ai.wu.ac.at/teaming-ai/extensible-event-stream-to-knowledge-graph>.

```

SELECT ?dataAssociation ?dataStoreReferenceName ?predicateToActivity ?activityName
WHERE {
  ?dataAssociation rdfs:subClassOf ?o .
  FILTER ( ?o = bboe:DataInputAssociation || ?o = bboe:DataOutputAssociation ) .
  ?dataAssociation ?predicateReferences ?reference .
  FILTER ( ?predicateReferences = bbo:has_sourceRef ||
           ?predicateReferences = bbo:has_targetRef ) .
  ?reference rdfs:subClassOf bboe:DataStoreReference .
  ?dataAssociation ?predicateToActivity ?activity .
  FILTER ( ?predicateToActivity = bboe:is_dataInputFor ||
           ?predicateToActivity = bboe:is_dataOutputFrom ) .
  ?activity bbo:name ?activityName .
  ?reference bbo:name ?dataStoreReferenceName .
}

```

dataAssociation	dataStoreReferenceName	predicateToActivity	activityName
tai:DataOutputAssociation_12b2glg	Machine Log Data	bboe:is_dataOutputFrom	Produce part
tai:DataOutputAssociation_17xegjx	Automatic quality checking result	bboe:is_dataOutputFrom	Check quality automatically
tai:DataOutputAssociation_1spmd24	Part information	bbe:is_dataOutputFrom	Save part information
tai:DataInputAssociation_0blyrx1	Automatic quality checking result	bboe:is_dataInputFor	Check if it is a reappearing defect
tai:DataInputAssociation_10dfmfx	Manual quality checking result	bboe:is_dataInputFor	Check if it is a reappearing defect
tai:DataOutputAssociation_0iuiqz3	Manual quality checking result	bboe:is_dataOutputFrom	Check quality manually
tai:DataOutputAssociation_12tyfc6	Machine parameters	bboe:is_dataOutputFrom	Produce a test part
tai:DataOutputAssociation_1tg1wtf	Quality result	bboe:is_dataOutputFrom	Check quality

**Fig. 5.** SPARQL query and result for (Q1) showing data flows between activities and data stores.

## References

1. Erasmus, J., Vanderfeesten, I., Traganos, K., Grefen, P.: Using business process models for the specification of manufacturing operations. *Comput. Ind.* **123**, 103297 (2020)
2. Abouzid, I., Saidi, R.: Proposal of BPMN extensions for modelling manufacturing processes. In: 5th International Conference on Optimization and Applications (ICOA), pp. 1–6. IEEE (2019)
3. Abramowicz, W., Filipowska, A., Kaczmarek, M., Kaczmarek, T.: Semantically enhanced business process modeling notation. In: *Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications*, pp. 259–275. IGI Global (2012)
4. Ahn, H., Chang, T.-W.: Measuring similarity for manufacturing process models. In: Moon, I., Lee, G.M., Park, J., Kiritsis, D., von Cieminski, G. (eds.) *APMS 2018*. IAICT, vol. 536, pp. 223–231. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-99707-0\\_28](https://doi.org/10.1007/978-3-319-99707-0_28)
5. Annane, A., Aussenac-Gilles, N., Kamel, M.: BBO: BPMN 2.0 based ontology for business process representation. In: *20th European Conference on Knowledge Management (ECKM 2019)*, vol. 1, pp. 49–59, Lisbon, Portugal, September 2019
6. 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
7. Buchgeher, G., Gabauer, D., Martinez-Gil, J., Ehlringer, L.: Knowledge graphs in manufacturing and production: a systematic literature review. *IEEE Access* **9**, 55537–55554 (2021)

8. Calvanese, D., Kalayci, T.E., Montali, M., Tinella, S.: Ontology-based data access for extracting event logs from legacy data: the *onprom* tool and methodology. In: Abramowicz, W. (ed.) BIS 2017. LNBIP, vol. 288, pp. 220–236. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59336-4\\_16](https://doi.org/10.1007/978-3-319-59336-4_16)
9. Cinpoeru, M., Ghiran, A.-M., Harkai, A., Buchmann, R.A., Karagiannis, D.: Model-driven context configuration in business process management systems: an approach based on knowledge graphs. In: Pańkowska, M., Sandkuhl, K. (eds.) BIR 2019. LNBIP, vol. 365, pp. 189–203. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-31143-8\\_14](https://doi.org/10.1007/978-3-030-31143-8_14)
10. Clark, J., DeRose, S.: XML path language (XPath) version 1.0. W3C recommendation, W3C, November 1999. <https://www.w3.org/TR/1999/REC-xpath-19991116/>
11. Corea, C., Fellmann, M., Delfmann, P.: Ontology-based process modelling - will we live to see it? In: Ghose, A., Horkoff, J., Silva Souza, V.E., Parsons, J., Evermann, J. (eds.) ER 2021. LNCS, vol. 13011, pp. 36–46. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-89022-3\\_4](https://doi.org/10.1007/978-3-030-89022-3_4)
12. Cyganiak, R., Sundara, S., Das, S.: R2RML: RDB to RDF mapping language. W3C recommendation, W3C, September 2012. <https://www.w3.org/TR/2012/REC-r2rml-20120927/>
13. Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 Concepts and Abstract Syntax. W3c recommendation, World Wide Web Consortium, 25 February 2014. <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>
14. Das, S., Sundara, S., Cyganiak, R.: R2RML: RDB to RDF mapping language. W3C recommendation, W3C, September 2012. <https://www.w3.org/TR/2012/REC-r2rml-20120927/>
15. Erasmus, J., Vanderfeesten, I., Traganos, K., Grefen, P.: The case for unified process management in smart manufacturing. In: 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC), pp. 218–227 (2018)
16. Grangel-González, I., Halilaj, L., Vidal, M.-E., Rana, O., Lohmann, S., Auer, S., Müller, A.W.: Knowledge graphs for semantically integrating cyber-physical systems. In: Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., Wagner, R.R. (eds.) DEXA 2018. LNCS, vol. 11029, pp. 184–199. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-98809-2\\_12](https://doi.org/10.1007/978-3-319-98809-2_12)
17. Hoch, T., et al.: Teaming.AI: enabling human-AI teaming intelligence in manufacturing. In: Proceedings of Interoperability for Enterprise Systems and Applications Workshops: AI Beyond Efficiency: Interoperability towards Industry 5.0. Springer, Valencia (2022)
18. Hogan, A., et al.: Knowledge graphs. *ACM Comput. Surv. (CSUR)* **54**(4), 1–37 (2021)
19. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business process modeling: current issues and future challenges. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 501–514. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-02144-2\\_39](https://doi.org/10.1007/978-3-642-02144-2_39)
20. 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)
21. Kchaou, M., Khlif, W., Gargouri, F., Mahfoudh, M.: Transformation of BPMN model into an OWL2 ontology. In: International Conference on Evaluation of Novel Approaches to Software Engineering, pp. 380–388. Virtual Event, April 2021

22. Klingenberg, C.O., Borges, M.A.V., Antunes Jr., J.A.V.: Industry 4.0 as a data-driven paradigm: a systematic literature review on technologies. *J. Manuf. Technol. Manag.* (2019)
23. Malyshev, S., Krötzsch, M., González, L., Gonsior, J., Bielefeldt, A.: Getting the most out of Wikidata: semantic technology usage in Wikipedia’s knowledge graph. In: *International Semantic Web Conference*, pp. 376–394, Monterey, California, USA, October 2018
24. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* **62**(8), 36–43 (2019)
25. Business Process Model and Notation (BPMN) 2.0 specification (2011). <https://www.omg.org/spec/BPMN/2.0/PDF>, version 2
26. Patel, P., Ali, M.I., Sheth, A.: From raw data to smart manufacturing: AI and semantic web of things for industry 4.0. *IEEE Intell. Syst.* **33**(4), 79–86 (2018)
27. Polyvyanyy, A., Pika, A., ter Hofstede, A.H.: Scenario-based process querying for compliance, reuse, and standardization. *Inf. Syst.* **93**, 101563 (2020)
28. Riehle, D.M., Jannaber, S., Delfmann, P., Thomas, O., Becker, J.: Automatically annotating business process models with ontology concepts at design-time. In: de Cesare, S., Frank, U. (eds.) *ER 2017*. LNCS, vol. 10651, pp. 177–186. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-70625-2\\_17](https://doi.org/10.1007/978-3-319-70625-2_17)
29. Rivas, A., Grangel-González, I., Collarana, D., Lehmann, J., Vidal, M.-E.: Unveiling relations in the Industry 4.0 standards landscape based on knowledge graph embeddings. In: Hartmann, S., Küng, J., Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) *DEXA 2020*. LNCS, vol. 12392, pp. 179–194. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-59051-2\\_12](https://doi.org/10.1007/978-3-030-59051-2_12)
30. Roy, S., Dayan, G.S., Devaraja Holla, V.: Modeling industrial business processes for querying and retrieving using OWL+SWRL. In: Panetto, H., Debruyne, C., Proper, H.A., Ardagna, C.A., Roman, D., Meersman, R. (eds.) *OTM 2018*. LNCS, vol. 11230, pp. 516–536. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-02671-4\\_31](https://doi.org/10.1007/978-3-030-02671-4_31)
31. Scheer, A.W., Thomas, O., Adam, O.: *Process Modeling using Event-Driven Process Chains*, Chap. 6, pp. 119–145. Wiley, New York (2005)
32. Schneider, P.: Managerial challenges of industry 4.0: an empirically backed research agenda for a nascent field. *Rev. Manag. Sci.* **12**(3), 803–848 (2018)
33. Muehlen, M., Recker, J.: How much language is enough? Theoretical and practical use of the business process modeling notation. In: *Seminal Contributions to Information Systems Engineering*. LNCS, pp. 429–443. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36926-1\\_35](https://doi.org/10.1007/978-3-642-36926-1_35)