Knowledge Graphs in Digital Twins for AI in Production

Pieter Lietaert(✉), Bart Meyers, Johan Van Noten, Joren Sips, and Klaas Gadeyne

Flanders Make, Gaston Geenslaan 8, 3100 Heverlee, Belgium
pieter.lietaert@flandersmake.be

Abstract. AI is increasingly penetrating the production industry. Today, however, AI is still used in a limited way in a production environment, often focusing on a single production step and using out-of-the-box AI algorithms. AI models that use information spanning a complete production line and even larger parts of the product lifecycle could add significant value for production companies. In this paper, we suggest a digital twin architecture to support the complete AI lifecycle (discovering correlations, learning, deploying and validating), based on a knowledge graph that centralizes all information. We show how this digital twin could ease information access to different heterogenous data sources and pose opportunities for a wider application of AI in production industry. We illustrate this approach using a simplified industrial example of a compressor housing production, leading to preliminary results that show how a data scientist can efficiently access, through the knowledge graph, all necessary data for the creation of an AI model.

Keywords: Digital twin · Knowledge graph · Data architecture · AI in production

1 Introduction

With the production (=manufacturing and assembly) industry moving towards Industry 4.0, a large amount of information is recorded and collected by smart and interconnected Cyber-Physical Production Systems (CPPS). This information could and should be leveraged to learn from the past and from similar cases using Artificial Intelligence (AI) systems, where we use the definition of AI in [1] in this paper. AI systems can support or automate decisions, such as: intelligently pick product samples for quality inspection, optimize product and production performance, reduce the number of required iterations for tuning the machine settings in case of a production line changeover, to increase the sustainability of the product, etc. The enormous potential of AI in production has been valued by Accenture as 3,7 trillion USD by 2035 [2]. A Gartner study indicates that the AI transformation in production has started, as AI implementations grew by 37% during 2018, and by 270% over the last four years [3].

Today, production companies typically use AI only by employing out-of-the-box algorithms, in a singled-out production step, such as image recognition for quality control

Published by Springer Nature Switzerland AG 2021
A. Dolgui et al. (Eds.): APMS 2021, IFIP AICT 630, pp. 249–257, 2021.
https://doi.org/10.1007/978-3-030-85874-2_26

(defect detection). Such AI algorithms are typically black-box, as those also commonly used in other domains, by companies like GAFA (Google, Apple, Facebook and Amazon). These algorithms require large amounts of data for training and validation, acquired by measuring targeted values relevant to the production step under consideration.

A largely unsolved challenge, however, is how AI algorithms can use information over multiple production steps, possibly even the entire product lifecycle, from design to product use. This challenge becomes even more apparent when considering the trend towards high mix, low volume (HMLV) production, with smaller series and more operator involvement for better flexibility. In this setting, the amount of data gathered for one variant is typically limited and too diverse to apply out-of-the-box AI algorithms. In addition, AI systems are only considered trustworthy [4] in the context of production if it is possible to explain to operators why a certain suggestion is made, demanding a more transparent AI approach.

One particular challenge that a data scientist faces when creating reliable and transparent models in this context, is the ability to access all required and relevant information (and preferentially not more) over the complete lifecycle of AI design, i.e. while finding correlations, learning, deployment, execution and validation. Information spanning larger parts of the product lifecycle typically requires access to multiple, heterogeneous data sources, including relational and non-relational (fi. time-series) databases, simulations, web APIs, user manuals, etc. The case of HMLV production further intensifies the need for gathering data from diverse sources, since model reliability can drastically be improved by supplementing the limited amount of measurement data with additional information, such as domain expert knowledge, operator experience and physics models. Currently, searching for the correct information in a typical industrial context causes prohibitively expensive overhead to the data scientist who is trying to find new correlations and models that could add a lot of value to the company. We will also investigate these challenges in the ICT-38–2020 ASSISTANT project.

In this paper, we consider this information access problem and suggest a data architecture centered around a digital twin that is based on a domain-wide knowledge graph. In Sect. 2 we describe this approach, illustrating it with an example of compressor housing manufacturing. In Sect. 3, we describe some experiments, illustrating the type of techniques that would enable easy data access in a knowledge graph centered architecture. In the final section we summarize our findings and present the next steps to take for its realization.

2 Approach

In an industrial context, a data scientist needs to (1) gather data over a vast set of heterogeneous data sources and (2) gather knowledge about the many production processes that exist in the company. In order to support this data scientist, we suggest the use of a digital twin built around a formal knowledge graph. Here, we use the term digital twin to indicate the central part of the data architecture in the production company, storing and providing access to all offline and online data. A knowledge graph, as the name suggests, organizes the information in a graph-like, and thus interlinked way. It has been made famous by initiatives, like, for example, DBpedia [5] and Google Knowledge Graph

[6]. In an industrial context, the knowledge graph can be used to capture and formalize the available, domain-wide meta-data, to formalize implicit expert knowledge and to provide the central access point to retrieve information. With regards to legacy data storage and scalability, typically, the knowledge graph should not contain large amounts of actual data, such as time series individuals. Instead, it should reference access to this data and therefore rely on meta-data.

The knowledge graph serves three main goals:

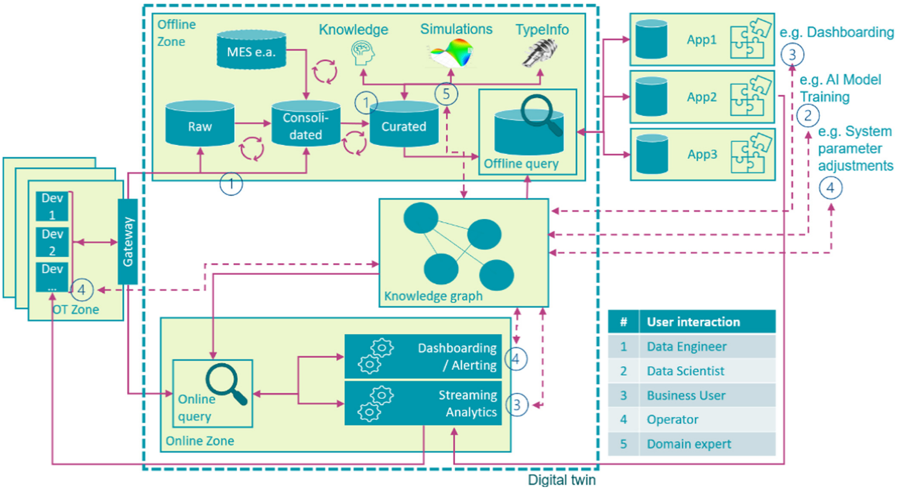
1. create a common vocabulary across the multiple disciplines in production,
2. facilitate knowledge search, capture and creation, i.e. identification of domain concepts and (new) relations among these concepts, and,
3. facilitate data search, i.e. connecting the domain concepts to the set of heterogeneous data sources.

The first goal will enable involved parties throughout the company to better find and understand data available to them and to more easily access the data they need. There has been research on how to describe production domain knowledge in a formal way [7–12]. However, many of the schemes lack expressiveness in certain areas, like ways to describe operator knowledge, or lack the ability to directly include known physics relations, models or constraints. Furthermore, although there are examples towards integrating formalized schemes in industrial applications, see e.g. [13, 14], there does not seem to be widespread adaptation yet.

The second goal involves creating knowledge from that data. This is not restricted to linking raw data, i.e. the knowledge graph should allow to link domain concepts and data sources themselves, e.g. linking types, algorithms, models and simulations. With the proper tools, different users can add new concepts and links to the knowledge graph, increasing knowledge within the company over time. There are a number of technologies available to realize linked data in the knowledge graph. For example, the World Wide Web Consortium (W3C) introduced the Resource Description Framework (RDF) [17] in 1996 as a way to describe linked data. Later it added reasoning rules, like for example expressed in the Web Ontology Language (OWL), that allow to find new links more easily in an algorithmic way. Another one is metamodeling [20], which allows the precise description of the types, relationships and constraints for a domain.

When using a knowledge graph, the third goal, i.e. data access, can be facilitated through semantic queries, see e.g. [15, 16, 21, 22]. The semantic query ensures that users can ask for exactly the data they need, rather than collecting data from different data sources and combining (joining) data manually. Furthermore, the user does not have to be concerned with the actual data sources that are being queried, if the central knowledge graph enables a performant connection between the contained concepts and the actual data sources. Data federation through a central, semantic query point is already possible using integrated software like the Ontotext platform [18], Timbr [19] and many others. However, it seems that this type of software has not penetrated many production company workflows yet.

This leads us to suggest the generic, high-level, digital twin centered architecture that can be seen in Fig. 1. Here, we take typical data architecture practice in industry, and add the idea of a knowledge graph based digital twin, creating an architecture that can support the data scientists in all phases of AI model creation. On the input side on the left, different devices are connected to a gateway, that either sends the data to storage, or can be directly queried in the case of online applications, like dashboarding and streaming analytics. In case of storage, in the offline zone on top, the incoming machine data is typically consolidated together with data coming from other company sources such as order processing or Manufacturing Execution Systems (MES). Next, a curated data zone should be created, with cleaned data. Moreover, it combines data from different sources throughout the product lifecycle, such as simulation data, type info, or operator knowledge. Next, different stakeholders such as business users and data scientists can use the digital twin to discover existing and add new information and knowledge (dashed arrows) and, finally, access data through queries facilitated - ideally automatically generated - by the digital twin (full arrow between knowledge graph and offline query). Similarly, for people using online applications, the digital twin serves as a reference to the concepts that are important to the application and facilitates online querying of the data provided by the gateway. Note that this architecture can also serve as a starting point for integration of existing data federation tools in the production company.



#	User interaction
1	Data Engineer
2	Data Scientist
3	Business User
4	Operator
5	Domain expert

Fig. 1. High-level overview example of architecture with digital twin based on central knowledge graph.

The knowledge graph centered architecture should enable the three goals mentioned before. To illustrate this, consider the following example of a compressor housing that undergoes a series of CNC-controlled machining operations. This example is a very simplified version of one of the industrial partners in the ICT-38–2020 ASSISTANT project. In Fig. 2, a number of relevant concepts relating to this part of the production process have been expressed in a high-level, abstract view of a knowledge graph (in blue).

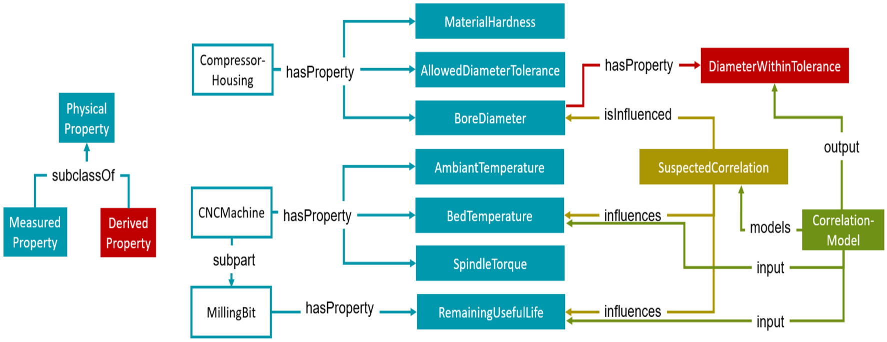


Fig. 2. High-level, abstract view of a knowledge graph for the compressor housing example. (Color figure online)

Notice how the overview on the left contains the concepts of a measured physical property and derived physical property. This allows different stakeholders throughout the company to relate their interpretation of physical quantities, like material hardness and temperature, to concepts in the knowledge graph, easing the interpretation of data on these quantities from different sources, like measurements or physics models. Furthermore, concepts like ‘suspected correlation’ in the knowledge graph allow to express relations that are based on operator experience, rather than using pure data.

In order to see how the knowledge graph can be used to create new knowledge, we can apply it to the data scientist’s workflow for creating an AI model. We can subdivide this workflow in 7 steps, depicted in Fig. 3.

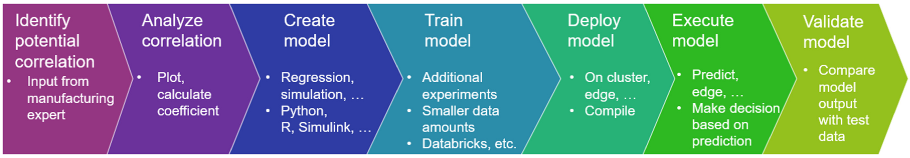


Fig. 3. Overview of the seven steps a data scientist can take to create an AI model.

Assume the accuracy of the housing’s rotor bore diameter is one of the main influencers of the efficiency of the compressor. A data scientist is tasked to create a model that predicts, during production, whether the diameter will be in tolerance, and he takes the following steps.

1. First, he asks a domain expert, who is familiar with the production process, for influencing factors of the bore diameter. The domain expert, based on his experience of the production process, adds a ‘SuspectedCorrelation’ to the knowledge graph, which relates different production properties as factors influencing the bore diameter, see Fig. 2 (yellow).

2. Then, the data scientist consults this correlation concept, investigates the influencing factors, and, for example, makes a plot of the data and can possibly turn the suspected correlation into a proven correlation.
3. Next, he creates a model that computes the precision of the diameter during the production of a specific housing, given the circumstantial evidence collected online. He adds a reference to this model to the knowledge graph, as can be seen in Fig. 2 (green).
4. To train the model, he accesses the knowledge graph to get training data. If not enough data is available (e.g. find out if sufficient 'BedTemperature' time series data was logged), he asks for more experiments.
5. Next, the data scientist deploys the model on an edge device to allow live computation of the diameter. He adds meta-data of this deployment, e.g. on which device it is running, to the knowledge graph (for example as a property of the model). Since the model is referenced like this, it is easy to find and access by other stakeholders in the future, e.g. a control engineer who wants to use the model in a smart controller.
6. He also introduces a 'DiameterWithinTolerance' property, see Fig. 2 (red), as the output of the model. When actual computations are made by the model, the property references this new data in the knowledge graph. He connects this property to the concept of the bore diameter, so that it can easily be found in the future when investigating the bore diameter.
7. Finally, the data scientist validates the model based on input from an operator. He can use the knowledge graph to quickly find the measurements from the operator, needed for validation.

Of course, the knowledge graph should be able to link the concepts in the graph to the correct data. This is also apparent in the above described 7-step process. For example, in step 2, to be able to plot the data, and in step 4, to be able to train the data, the data scientist should be able to easily access the individual data elements that where linked as influencing factors of the diameter, such as time series temperature data, measured diameters, or tool information on remaining life. Furthermore, once the model is trained and deployed, other people, also later in time, should be able to find such models in the knowledge graph, run them on new input data and get the output data values. Note that the knowledge graph centered architecture is not only suited for AI model creation, but also provides the basis for, more generally, access to all knowledge gathered and contained in the company.

In the next section, we illustrate how interaction with the knowledge graph could look like, using semantic querying with two different techniques.

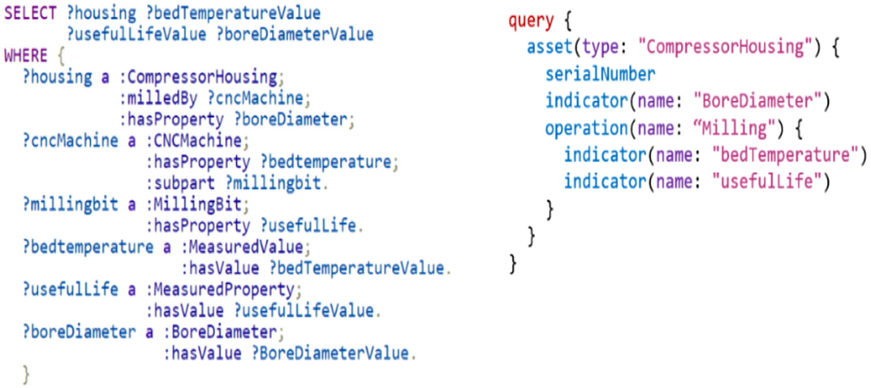
3 Querying Examples

In this section, we present two different implementations for data access through a digital twin based on the knowledge graph in Fig. 2: (1) a knowledge graph represented using the W3C RDF triples format accessed through SPARQL queries to retrieve the data from a relational database, or, (2) a meta-model style knowledge graph accessed through GraphQL queries, where a GraphQL schema and implementation provides access to the data stored in the knowledge graph.

Presume the data scientist wants to investigate all influencing factors of the bore diameter, in the “offline zone”, as in step 2 in the previous section. In Fig. 4, on the left, you can see a SPARQL query that would result in actual data values for all of the influencing factors and the bore diameter. The SPARQL query allows an intuitive way of accessing related data. In this example, all compressor housings are linked to the bed temperature of the CNC machine they were milled on, to the remaining life of the milling bit that was used and to the bore diameter value that resulted from the milling process. We can use Ontop [21] to perform this SPARQL query over data stored in a relational database. This avoids the user having to get familiar with the technical database schema and, rather, allows users to ask questions over a knowledge graph storing concepts like ‘CompressorHousing’, ‘CNCMachine’, connected by properties such as ‘milledBy’.

On the right, an equivalent query in GraphQL is shown. In the GraphQL case, data is also accessed through intuitive connections expressed in the GraphQL schema, such as an asset of type ‘CompressorHousing’ having properties like serial number and operations. Technical data access, e.g. using queries directly to the relational database storing the data, is again avoided by translating the GraphQL query through, in this case, a custom API.

Both examples show data access through a central knowledge graph, using two different technologies, avoiding the requirement of an often complex, technical understanding of where and how the data is stored and instead employing intuitive concepts contained in the knowledge graph to get the required data.



```
SELECT ?housing ?bedTemperatureValue
       ?usefulLifeValue ?boreDiameterValue
WHERE {
  ?housing a :CompressorHousing;
           :milledBy ?cncMachine;
           :hasProperty ?boreDiameter;
  ?cncMachine a :CNCMachine;
              :hasProperty ?bedtemperature;
              :subpart ?millingbit.
  ?millingbit a :MillingBit;
              :hasProperty ?usefulLife.
  ?bedtemperature a :MeasuredValue;
                  :hasValue ?bedTemperatureValue.
  ?usefulLife a :MeasuredProperty;
              :hasValue ?usefulLifeValue.
  ?boreDiameter a :BoreDiameter;
                :hasValue ?BoreDiameterValue.
  }
```

```
query {
  asset(type: "CompressorHousing") {
    serialNumber
    indicator(name: "BoreDiameter")
    operation(name: “Milling") {
      indicator(name: "bedTemperature")
      indicator(name: "usefulLife")
    }
  }
}
```

Fig. 4. Examples of two different queries on the knowledge graph in Fig. 2.

4 Conclusion

We identify access of data and knowledge as a main bottleneck for manufacturing companies to apply AI solutions. To address this, we investigated the use of a knowledge graph that can be queried, and an architecture to apply the knowledge graph in a manufacturing context. We illustrated how the knowledge graph can support the data scientist in accessing information from heterogeneous data sources, including expert knowledge, throughout the complete AI lifecycle. A small query example showed how this can be applied in practice and how the knowledge graph facilitates efficient data access for the data scientist.

Three challenges remain before being able to apply this approach successfully in a production context.

First, the domain concepts in the knowledge graph should have the proper expressiveness in order to properly add less tangible information, such as operator experience, correlations, models and uncertainty.

Second, although the information access through querying was illustrated with two examples, we are still in the process of validating, together with production companies, which approach is best suited in the context of querying information from the knowledge graph based digital twin for AI.

Finally, the data architecture that was presented, showed data querying of the offline and online data sources as separate steps. It is not clear yet what the best practices are to link the information in the knowledge graph to the data sources. We will investigate these challenges in the ICT-38–2020 ASSISTANT project.

Acknowledgments. This research was supported by Flanders Make, the strategic research center for the manufacturing industry. This paper was partially funded by the DTDesign ICON (Flanders Innovation & Entrepreneurship FM/ICON :: HBC.2019.0079) project. We would also like to acknowledge the European Commission for funding through the ASSISTANT project, grant number 101000165.

References

1. A definition of Artificial Intelligence: main capabilities and scientific disciplines. https://digital-strategy.ec.europa.eu/en/library/definition-artificial-intelligence-main-capabilities-and-scientific-disciplines. Accessed 18 June 2021
2. Manufacturing the future. https://www.accenture.com/_acnmedia/pdf-74/accenture-pov-manufacturing-digital-final.pdf. Accessed 10 Mar 2021
3. Gartner Survey of More Than 3,000 CIOs Reveals That Enterprises Are Entering the Third Era of IT. https://www.gartner.com/en/newsroom/press-releases/2018-10-16-gartner-survey-of-more-than-3000-cios-reveals-that-enterprises-are-entering-the-third-era-of-it. Accessed 10 Mar 2021
4. Ethics guidelines for trustworthy AI. https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai. Accessed 10 Mar 2021
5. DBpedia. https://wiki.dbpedia.org/. Accessed 10 Mar 2021
6. Introducing the knowledge graph: things not strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/. Accessed 10 Mar 2021
7. Gayathri, R., Uma, V.: Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: a survey. ICT Express **4**(2), 69–74 (2018)
8. Sampath Kumar, V., et al.: Ontologies for Industry 4.0. Knowl. Eng. Rev. **34**(17), 1–14 (2019)
9. Kourtis, G., Kavakli, E., Sakellariou, R.: A rule-based approach founded on description logics for Industry 4.0 smart factories. IEEE Trans. Ind. Inform. **15**(9), 4888–4899 (2019)
10. Giustozzi, F., Saunier, J., Zanni-Merk, C.: Context modeling for Industry 4.0: an ontology based approach. Procedia Comput. Sci. **126**, 675–684 (2018)

11. Cao, Q., Giustozzi, F., Zanni-Merk, C., De Bertrand de Beuvron, F., Reich, C.: Smart condition monitoring for Industry 4.0 manufacturing processes: an ontology-based approach. Cybern. Syst. **50**, 1–15 (2019)
12. Heng, Z., Utpal, R., Yung-Tsun, T.L.: Enriching analytics models with domain knowledge for smart manufacturing data analysis. Int. J. Prod. Res. **58**(20), 6399–6415 (2020)
13. Kalaycı, E.G., et al.: Semantic integration of bosch manufacturing data using virtual knowledge graphs. In: Pan, J.Z., et al. (eds.) ISWC 2020. LNCS, vol. 12507, pp. 464–481. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62466-8_29
14. Kharlamov, E., et al.: Ontology based data access in statoil. J. Web Semant. **44**, 3–36 (2017)
15. Kharlamov, E., et al.: Optique: towards OBDA systems for industry. In: Cimiano, P., Fernández, M., Lopez, V., Schlobach, S., Völker, J. (eds.) ESWC 2013. LNCS, vol. 7955, pp. 125–140. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41242-4_11
16. Grangel-Gonzalez, I., Halilaj, L., Coskun, G., Auer, S., Collarana, D., Hoffmeister, M.: Towards a semantic administrative shell for Industry 4.0 components. In: Proceedings - 2016 IEEE 10th International Conference on Semantic Computing, ICSC 2016, pp. 230–237 (2016)
17. RDF. https://www.w3.org/RDF/. Accessed 10 Mar 2021
18. Ontotext platform. https://www.ontotext.com/products/ontotext-platform/. Accessed 10 Mar 2021
19. Timbr. http://timbr.ai/platform/. Accessed 10 Mar 2021
20. Thomas, K.: Matters of (meta-)modeling. Softw. Syst. Model. **5**(4), 369–385 (2006). https://doi.org/10.1007/s10270-006-0017-9
21. Calvanese, D., et al.: Ontop: answering SPARQL queries over relational databases. Semant. Web **8**(3), 471–487 (2017)
22. Sequeda, J., Miranker, D.: Ultrawrap: SPARQL execution on relational data. J. Web Semant. **22**, 19–39 (2013)