SEA4DQ '22: 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things
November 17, 2022
Singapore, Singapore

Conference Sponsors:
SIGSOFT

RESEARCH-ARTICLE

Data quality as a microservice: an ontology and rule based approach for quality assurance of sensor data in manufacturing machines

JØRGEN STANG, DNV GL AS, Oslo, Norway

DIRK B WALTHER, DNV GL AS, Oslo, Norway

PER MYRSETH, DNV GL AS, Oslo, Norway

SEA4DQ 2022: Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (November 2022)
https://doi.org/10.1145/3549037.3561272
ISBN: 9781450394598

Data Quality as a Microservice: An Ontology and Rule Based Approach for Quality Assurance of Sensor Data in Manufacturing Machines

Jørgen Stang*
Dirk Walther*
Per Myrseth*
jorgen.stang@dnv.com
dirk.walther@dnv.com
per.myrseth@dnv.com
DNV
Oslo, Norway

ABSTRACT

The manufacturing industry is continuously looking for production improvements resulting in high quality production, reduced waste and competitive advantages. In this article, ontologies, semantic rule logic and microservices have been deployed to suggest a system for quality assurance of manufacturing machine data. The existing upper ontology for manufacturing service description has been used to define both the physical assets as well as the data quality requirements. The system is used to both operationalize data quality monitoring by semantic technology as well as enabling up-front modelling of data quality requirements. The approach is illustrated by a specific speed-feed case for manufacturing machines but could easily be extended to other manufacturing use-cases or even to other industries.

CCS CONCEPTS

• **Computer systems organization** → **Architectures**; • **Theory of computation** → *Semantics and reasoning*; • **Applied computing** → Industry and manufacturing.

KEYWORDS

Data Quality, Manufacturing Machines, Sensor Data, Microservices, Ontolologies, IoT

ACM Reference Format:
Jørgen Stang, Dirk Walther, and Per Myrseth. 2022. Data Quality as a Microservice: An Ontology and Rule Based Approach for Quality Assurance of Sensor Data in Manufacturing Machines. In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3549037.3561272

*All authors contributed equally to this research.

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11…$15.00
https://doi.org/10.1145/3549037.3561272

1 INTRODUCTION

Internet of Things (IoT) and Industry 4.0 have already enabled digital transformation of knowledge, processes and services in most industries. Advanced technology is both common-place and off-the-shelf, providing means for quality improvements and efficiency gains. In the manufacturing industry, machines and devices are equipped with sensors to monitor operational characteristics such as speed, heat, vibrations and more. These characteristics are subsequently used to detect deviations, failure modes, inconsistencies and other events adversely affecting production quality. Traditionally, this monitoring would rely on humans to detect by noise, visuals, touch or even smell. Considering that the cost of rework and waste is significant [24] the digital capabilities providing continuous monitoring and analysis of operations is a game changer. In this new reality, the quality of the sensor data used for monitoring and issue handling will be a critical success factor. If left unattended, sensors, just as the physical assets, will malfunction, drift, freeze, misalign or plainly break. Subsequently, this will yield wrong information which in turn can both abate failure detection as well as trigger costly uncalled-for operations. This paper outlines how ontologies based on the Manufacturing Service Description Language (MSDL) can be used with data quality requirements expressed by the Semantic Web Rule Language (SWRL) as model constraints. The specific data quality rules are implemented as a microservice that can be reused and deployed both across manufacturing machines as well as across domains.

The work described here is intended to contribute towards the ambition of zero defect manufacturing [13], where all process, product and data is monitored to ensure any deviations are captured and handled before production quality is affected.

2 OVERALL SYSTEM ARCHITECTURE

The described system uses a 3 layered architecture and each component is briefly described in the next sections. The main objective is to provide data quality monitoring by automatic reasoning (inference) for the manufacturing machine signals. Figure 1 shows the 3 layers; (1) model definition by ontology on top, (2) semantics rule logic in the middle and (3) the data quality service at the bottom. All three layers are required to efficiently represent:

- Ontology – Model constraints (e.g. machine has a feed);

- Rule – Semantic constraints (e.g. correlation coefficient should be more than 0.8 between feed and cutting speed for a single axis machines); and
- Metric – Data Quality as a Service (e.g. calculate correlation coefficient).

The following is a brief description of Machine Signal, Ontology, Rule, Metric and Result as shown in Figure 1.

Machine signals are received by event-hubs, streaming APIs, historical data loads or other integration methods. Usually signals are buffered to enable data quality metrics to be performed on more than a single datapoint. Even though some metrics are useful on single data points (say range and code list validation), most metrics will require a dataset as input. Also, performance issues will often debate operations on single datapoints.

The buffered dataset is uploaded to the ontology and validated according to the defined relations. This utilises the capabilities of the ontology to define complex models and assign a meaning to the relations itself. The particular ontology used here is described in a later section. The valid model can subsequently be queried by rules to determine compliance with requirements. Ontologies have reasoning capabilities and in addition we deploy a dedicated rule definition language such as SWRL [12], SHACL [16] or SPIN [15]. Rule languages will add query capabilities that is not provided directly by the ontology itself, however, the model semantics are utilised to provide "smart" rule execution, meaning, depending on the machine configuration, different rules and different rule parameters will be used.

The rule will in turn trigger a data quality metric. The metric could be compute intensive and rely on complex algorithms and is therefore implemented in compiled code that executes on scalable compute resources such as Kubernetes clusters [10], virtual or on-premise machines. The choice of compute resource will often rely on security issues and enterprise strategy for cloud or federation.

The data quality assessment result is subsequently used for alarms, notifications and trend analysis, and will ultimately drive the improvement processes to ensure that the data produced by the manufacturing machine sensors are fit for use and that operations are performed within acceptable risk.

The next sections will describe the 3 main components; Ontology, Rule and Metric in more detail.

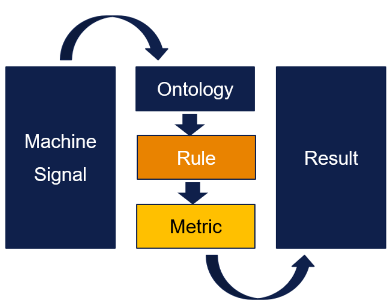


Figure 1: System Architecture

3 ONTOLOGY FOR MANUFACTURING

Ontologies are generally implemented to model complex relations in information models. Properly defined, ontologies can be used to both represent knowledge in the model as well as to provide inference capabilities (automatic reasoning) enabling advanced queries. Knowledge is represented by semantics that define the meaning of a relation, whereas inference is used to query the model by semantics. These capabilities makes ontologies a very powerful modelling and validation tool [11]. As a very simple example, in a manufacturing machine, an ontology can represent a carving machine with 1 tool which in turn can have spindle positions that move in 1-3 directions (x,y,z). The ontology can hence validate that the machine configuration is correct (1 tool and 1-3 directions), in addition, the inference mechanism can use the number of directions present to infer other characteristics. This example is elaborated further in a later section.

Well defined ontologies are a prerequisite to ensure scaling on performance and model complexity. Ontologies are commonly divided into layers which represent different levels of abstraction [7]. The top ontology, or upper ontology, will focus on abstract entities and the middle layer adds domain specific entities. Based on the upper and middle ontology, specialised entities and constructs can be added to provide custom implementations. This provides for an extensible and flexible modelling paradigm, at the same time, overlapping lower ontologies in any layer should be avoided. To that end, several industries have established fora for the development and exchange of common ontologies [14] [19] [23] [20]. The Manufacturing Service Description Language (MSDL) was first introduced in 2006 [2] and further revised in 2019 [5]. It was developed to support interoperability and advanced reasoning for manufacturing services within a Virtual Enterprise (VE) [2]. Except for mentions in recent PhD theses [18] [17] and several articles [5] [3], there is no evidence of extensive adoption of MSDL in the industry [4], however, currently, MSDL has been incorporated into the Industrial Ontologies Foundry (IOF) initiative and will probably as a consequence gain more traction. In this work we use MSDL as the upper ontology as the current version of MSDL is well suited for our purpose. By nature, ontologies evolve and are designed to be extended for special purposes. New versions of MSDL or other suitable upper ontologies for manufacturing could hence also be incorporated in the future.

As shown in Figure 2, MSDL divides the manufacturing service into supplier, process and resource. Machining capabilities is a resource which again includes physical components such as axes, tables, spindle and tool [28]. We also include the entities feed and clock as the relationship between cutting speed (spindle) and feed rate is an important driver for product quality [28] and, also, clock speed and synchronization is required for both analysis and monitoring. Figure 2 show the MSDL based ontology used in the work presented here.

4 ONTOLOGY RULE LOGIC EXPRESSIONS

Several formal constructs have been implemented to add rule logic capabilities to ontologies [12] [16] [15]. This is useful to express specific rules that is not supported by the ontology itself. The ontology described in the previous chapter and shown in Figure 2 can

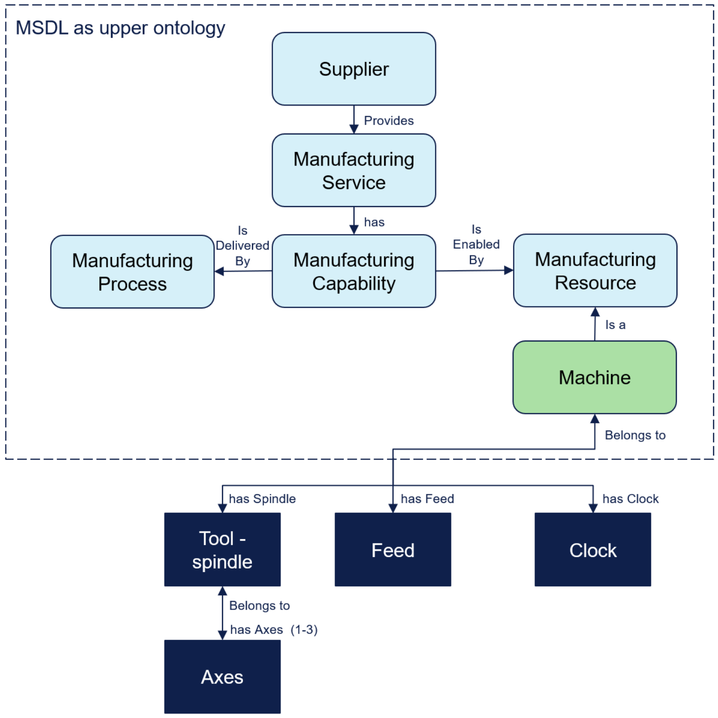


Figure 2: Ontology for manufacturing (MSDL) used as upper ontology

express that a machine has 1 feed, 1 spindle with 1-3 axes and 1 clock. However, rule logic expressions can be used to determine relationships between specific machine configurations and required query results. In the context of data quality, there could be different requirements to correlation coefficient between spindle speed and feed rate depending on the number of axes. For 1 axis, a typical drilling operation, the correlation could be linear, whereas for spatial operations with 3 axes the correlation could be weaker or even non-existent. Hence, rule logic can be expressed as

> *SELECT* rule *WHERE* machine hasFeed *COUNT* 1 *AND* hasSpindle *COUNT* 1 *AND* has Axes *COUNT* 3 *AND* result < 0.4

will return the appropriate data quality rule. The data quality rule will be executed by external service (DQaaS) and the result yields acceptance criteria for this particular data quality metric. The above rule illustrates how ontologies and rules can be layered to provide powerful querying capabilities.

Rule languages are currently more immature than ontologies and hence not standardised and formalized to the same extent. The above statement is only for illustration and do not adhere to a specific rule format. The specific number used as rule result threshold (0.4) is explained later.

5 DATA QUALITY AS A MICROSERVICE

The term data quality as a service (DQaaS) denotes an existing library of data quality metrics that can be accessed as a cloud service or it can be deployed to local clusters [6] [13]. The label 'micro' simply indicates the service is stateless, specialised and will require some level of orchestration by an API-gateway or client applications. The DQaaS API provide access to methods as endpoints and the CPU can be provided by Kubernetes clusters, virtual machines or on-premise servers. The service provides the bottom layer of the architecture where compute intensive operations are performed. The service is implemented in Python using Pandas, Numpy, Great Expectations and other standard modules. The API complies to OpenAPI 3.0 [25] and is implemented with FastAPI [27]. The data quality metrics are predominantly geared towards IoT time series data in the format *timestamp — signal — value* which easily adopts itself to manufacturing machine signals. Some example metrics are shown in Figure 3. Duplicates, missing values, invalid values, invalid distributions and other anomalies are covered by the service, currently there are approximately 20 rules available. The rules are intended to monitor sensors for anomalies such as miscalibration, drift, freeze, downtime, clock-synchronization, noise, malfunction and others. Some metrics are defined in more detail in a later section. The following data quality issues are shown in Figure 3: Time collision (duplicate timestamps for same signal), outside range (where range is defined from min to max), Rate of Change (RoC), missing data (values or records) and drift.

Data quality is often defined by data quality frameworks [22] [8] and there are also dedicated standards such as ISO 8000 [1]. Typically the frameworks will categorize and suggest specific metrics definitions. ISO 8000 offers the distinction between syntax, semantic and pragmatic data quality, meaning format errors (wrong data type), invalid data (according to real world asset) and use case dependent respectively. The DQaaS is predominantly concerned with the semantic category and the metrics are based on common issues typically encountered for time series data [21] [9]. The pragmatic, or use case dependent category, is often used for different system configurations, for example, use cases involving analytics (prediction or trending) will require data quality measured at a high resolution. On the other hand, detecting failure modes with long *p–f* intervals (time from detection of *potential failure* to *–failure* happens) could require measurements at lower resolutions. Hence, it should be noted that data that have good quality for one use-case (say long p–f intervals) could be unfit for other use cases (say predictive analytics).

In addition to the broad categories syntactic, semantic and pragmatic, ISO 8000 also offers more detailed data quality characteristics and data quality anomalies. In the use case presented in a later section, we look at the characteristic called *consistency* (between feed rate and spindle speed) and the resulting anomaly *drift*. The consistency is calculated as the correlation coefficient for the related sensor signals. Drift occurs when related sensor signals experience increasing deviations, this is also shown graphically in Figure 3.

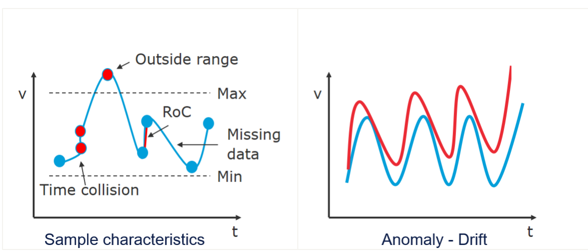


Figure 3: Sample data quality metrics for time series data

6 CONCEPTUAL MODEL

The overall architecture shown previously can be expanded to a more detailed conceptual model. The MSDL upper ontology is extended to include entities that will serve as input to the data quality service. The rule semantics connects related entities and the relevant rules defined in the data quality service. In Figure 4 the manufacturing machine is modelled by asset components, however there is no classification or modelling of the data quality rules. This asset-in-focus approach is commonly used to support digital twins and data interoperability, meaning there should be a consensus on how to represent the physical asset digitally. When this consensus is reached, the same unified model can be reused across machines as well as enterprises. The data quality metrics will then be called based on asset configuration and data quality requirements defined in the model. If a machine has a clock it should have a frequency and the frequency should be according to a given value. If the requirement is 1 Hz frequency and the monitored frequency is higher, this should trigger a notification that defines an action to be taken to mitigate the issue.

Alternatively, the ontology could emphasize on the data quality rules rather than the asset, focusing on data integration rather than interoperability. Any proprietary model could connect to the rule ontology by mapping individual terms to a rule vocabulary. This alleviates the requirements for a common information model and relies on the rule classification to define correct semantics. This approach has not been pursued further here but could prove useful for a use cases where there are complex requirements or regulations that should be applied to disparate data sources. The regulations (say GDPR) can then be modelled by the means described here and subsequently be applied to proprietary systems (say CRM).

The MSDL based ontology has attributes that describes physical features such as clock frequencies, feed rates and spindle positions. These attributes are managed by the query mechanism to trigger data quality rules. The data quality rules shown here are not exhaustive but represents commonly used metrics:

- Noise – Measures deviation between values of same attribute with a sideways shift, random deviations indicates noise in signal
- Frequency – Calculates lag between sorted timestamps and compares to requirement
- Duplicates – Identical timestamp for same signal
- Range/Rate of change – According to defined min-max values / according to allowed rate of change
- Deviation – Allowed difference between data points
- Distributions – Statistical distribution requirements such as normal, chi-square, Smirnov and others
- Correlations – Calculates correlation coefficient for related attribute series

7 USE CASE

The implemented use case is shown in the below figure. The main motivation for looking at the spindle and the feed-rate is the effect any mis-configuration of these parameters will have on the end product. Surface and finishing quality will deteriorate significantly if the material is fed out-of-sync with the cutting speed, tools can be damaged and material is wasted [28]. Therefore, careful monitoring of these critical parameters is required, and, subsequently the data quality should also be monitored. As mentioned in a previous section we will focus on the data quality characteristic *consistency* and the data quality anomaly *drift* as defined by ISO 8000. Consistency is calculated as the correlation coefficient for the two related sensors, feed rate and spindle speed. The relationship between feed rate and spindle speed will depend on material type, cutting axes and others [26]. The below formulae defines the mathematical relationships between cutting speed, spindle speed and feed [28]:

$$\text{Cutting speed: } V_c = \frac{\pi \ x \ D \ x \ n}{1000}$$
$$\text{Spindle speed: } n = V_c - \pi - D \ x \ 1000$$
$$\text{Feed: } V_f = n \ x \ f_z \ x \ Z$$

where D is the spindle diameter, f_z is feed per tooth and Z is number of flutes or teeth.

In this case it suffices to state that for any given machine configuration the relationship is constant. In addition, the ontology will yield model cardinality, as an example, the feed-speed correlation

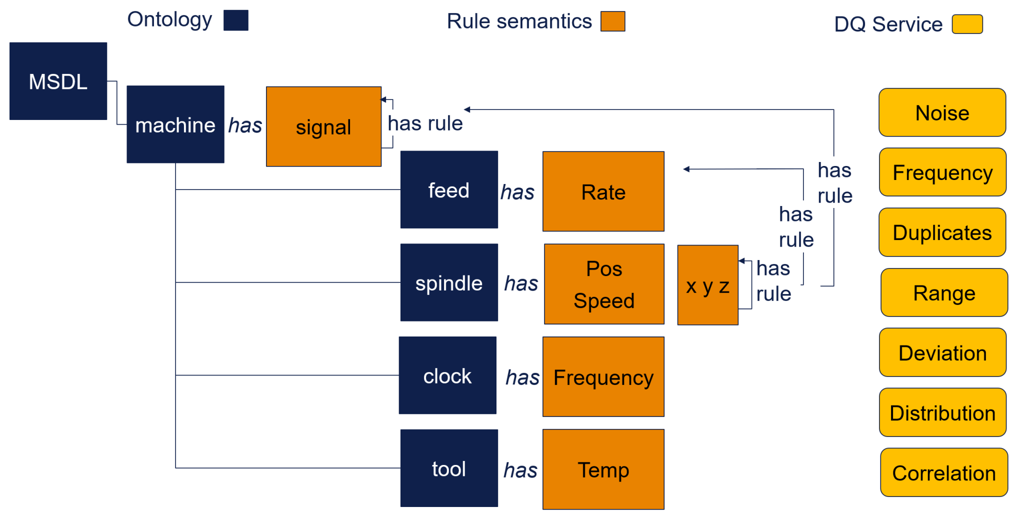


Figure 4: Conceptual model

depends on the number of axes and hence the automatic reasoning capabilities will handle correlation rules for different spatial carving capabilities (drilling, surface, moulding). Figure 5 show the speed-feed correlation for a moulding machine with spatial spindle positions (x,y and z) during two time intervals, t1 and t2. As shown, the feed will have different correlation to the 3 axes during the moulding process. The spindle z-position has high correlation in the first period t1 (0.4 from 07:38:30 to 07:39:00) whereas both z and x position has high values for the last period t2 (0.3 and 0.36 respectively from 07:40:00 to 07:41:00). If we assume the data used for the analysis have good quality, this can be used to set the threshold to detect any sensor data quality issues for the relevant sensors. The actual threshold value can be set up in a number of ways, here we simply say the squared value for the correlation coefficient for all axes should be above 0.4 (measured values are 0.42 and 0.48 for the two periods shown in the figure). This could be an oversimplification, also, we do not know if the data has already drifted or if there are any other data quality issues in the sample data, but it will suffice as an illustration for this use case.

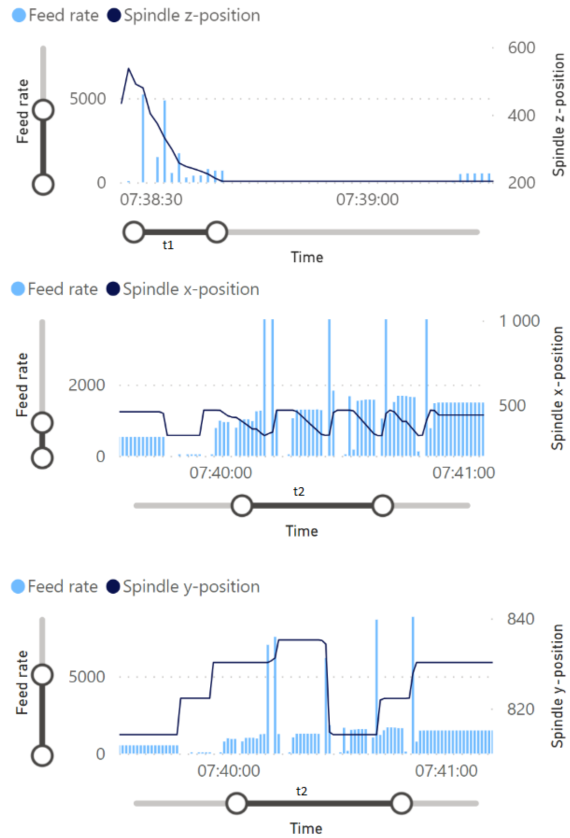
In addition to speed-feed correlation, the use case also includes rules for machine counter, status, clock frequency and valid ranges. The following is a list of the relevant rules, label in italic refers to Figure 6:

- *api:rule corr > 0.4* — feed/speed correlation coefficient should be above this value
- *api:rule <min ,max>* — Range for valid values, between min and max
- *api:rule <0,inf>* — Values should be above 0
- *api:rule f=1 Hz* — Timestamp should have this frequency
- *api:rule >0 then ON* — If speed is above zero then machine status should be ON
- *api:rule [ON,OFF]* — Machine status should be ON or OFF
- *api:rule n+1>n* — Machine running counter should always be increasing

The purpose of the data quality rules is to detect anomalies in the data and this should trigger a root cause analysis to define activities to support continuous improvement. The above list is not exhaustive and domain expertise should be used to define additional rules that can be added to the service and again triggered by the rules and constraints defined in the model. The use case also illustrates how the knowledge model (ontology) can be used in an operational setting with constraints and requirements. Data from manufacturing machines can be loaded into the ontology in real time and described mechanisms will continuously evaluate compliance to requirements. Also, this modelling approach ensures that data quality requirements are considered up front as part of modelling and design, and not as an ad hoc afterthought, which is often the case.

8 CONCLUSION

The Manufacturing Service Description Language (MSDL) have been used as a basis for a tentative extension to express data quality requirements for a simple manufacturing machine with clock, feed and spindle. A generic data quality service for sensor data can be used to calculate the data quality requirements based on semantic rule expressions such as SWRL. The data quality service was implemented in Python based on Great Expectations and deployed as a microservice on a cloud platform. The work described here shows how ontologies can be used to both model the knowledge (terminology/structure/semantics) of the asset as well as defining requirements and constraints to the production process itself. The resulting regime provides quality assurance of complex assets, even digital twins, and will apply relevant data quality rules based on asset configurations. One example was used for illustration where the ontology will distinguish between linear carving (drilling) and spatial carving (moulding) and apply appropriate requirement to the correlation between machine status, clock frequency, spindle speed and feed rate.



Time t1	*Axis*	*Feed rate corr*
07:38:30 - 07:39:00	Spindle z-position	0.4
07:38:30 - 07:39:00	Spindle y-position	0.086
07:38:30 - 07:39:00	Spindle x-position	0.09
	Total (squared)	0.42

Time t2	*Axis*	*Feed rate corr*
07:40:00 - 07:41:00	Spindle z-position	0.3
07:40:00 - 07:41:00	Spindle y-position	0.14
07:40:00 - 07:41:00	Spindle x-position	0.36
	Total (squared)	0.48

Figure 5: Correlation values for feed rate and spindle positions for sample data for a moulding machine

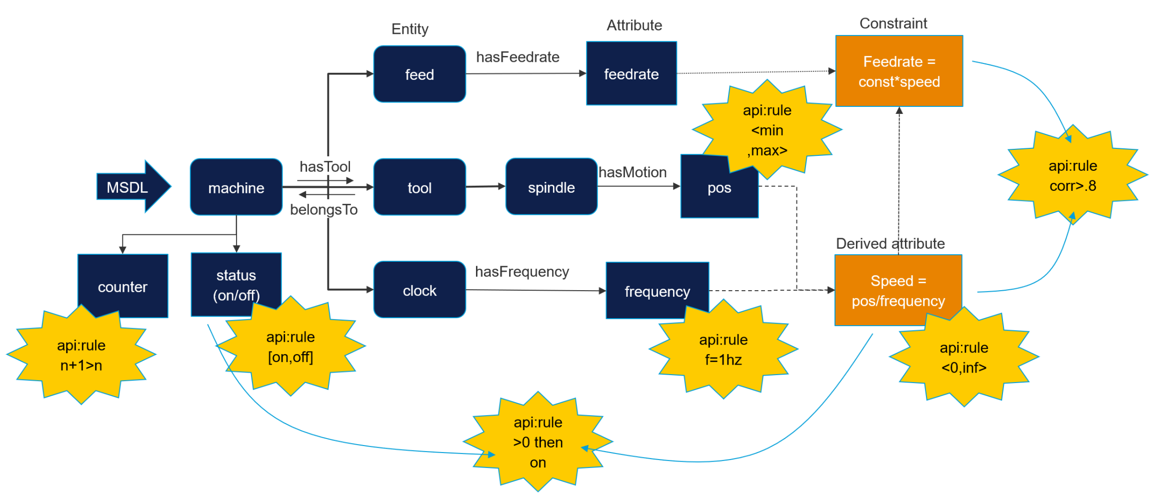


Figure 6: Use case

Jointly, the ontologies and the microservices can support both traditional model verification as well as process verification. This combination provides a manner to include and design for data quality during early design phases. Traditionally data quality tends to be left as an after-thought and implemented in an ad-hoc manner. Considering the significant importance of data quality in digital processes, the data quality requirements should be defined and implemented up-front.

Further work should look in to how the data quality result can be represented as an ontology itself and subsequently used as a driver for improvement activities and risk analysis. The ontology described here should be further expanded and formalised to support more advanced use-cases, also, the data quality service can be extended with additional rules. Ontologies will scale on both complexity and size, also, the microservice and cloud based architecture will deploy to any given compute capabilities.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 958357 and it is an initiative of the Factories-of-the-Future (FoF) Public Private Partnership.

REFERENCES

[1] ISO 8000-8. 2015. *Data quality — Part 8: Information and data quality: Concepts and measuring.* Retrieved June 23, 2022 from https://www.iso.org/standard/60805.html

[2] Farhad Ameri and Debasish Dutta. 2006. An upper ontology for manufacturing service description. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 42578. 651–661.

[3] Farhad Ameri, Christian McArthur, Bahram Asiabanpour, and Mohammad Hayasi. 2011. A Web-based Framework for Semantic Supplier Discovery for Discrete Part Manufacturing. In *Proceedings of NAMRI/SME*, Vol. 39.

[4] Farhad Ameri, Dusan Sormaz, Foivos Psarommatis, and Dimitris Kiritsis. 2022. Industrial ontologies for interoperability in agile and resilient manufacturing. In *Industrial Journal of Production Research*, Vol. 60.

[5] Damiano Arena, Farhad Ameri, and Dimitris Kiritsis. 2019. Skill Modelling for Digital Factories. In *IFIP International Conference on Advances in Production Management Systems (APMS).* 318–326.

[6] Veracity by DNV. 2022. *Veracity for Developers.* Retrieved June 23, 2022 from https://www.veracity.com/

[7] Oscar Cabrera, Xavier Franch, and Jordi Marco. 2015. A Middle-Level Ontology for Context Modelling. In *International Conference on Conceptual Modeling.*

[8] Corinna Cichy and Stefan Rass. 2019. An Overview of Data Quality Frameworks. *IEEE Access* 7 (2019), 24634–24648. https://doi.org/10.1109/ACCESS.2019.2899751

[9] DNV. 2017. Data Quality Assessment for Sensor Systems and Time-Series Data. In *DNV Report No. 2017-0058*, Vol. 1.01.

[10] Red Hat. 2020. *What is a Kubernetes cluster?* Retrieved June 23, 2022 from https://www.redhat.com/en/topics/containers/what-is-a-kubernetes-cluster

[11] Claudia Hess and Christop Schlieder. 2006. Ontology-based verification of core model conformity in conceptual modeling. In *Computers, Environment and Urban Systems*, Vol. 30.

[12] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. 2004. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved June 23, 2022 from https://www.w3.org/Submission/SWRL/

[13] InterQ. 2022. *Project for zero defect manufacturing.* Retrieved June 23, 2022 from https://interq-project.eu/

[14] IOF. 2022. *Industrial Ontologies Foundry (IOF) website.* Retrieved June 23, 2022 from https://www.industrialontologies.org/

[15] Holger Knublauch, James A. Hendler, and Kingsley Idehen. 2011. *SPIN - Overview and Motivation.* Retrieved June 23, 2022 from https://www.w3.org/Submission/spin-overview/

[16] Holger Knublauch and Dimitris Kontokostas. 2017. *Shapes Constraint Language.* Retrieved June 23, 2022 from https://www.w3.org/TR/shacl/

[17] Roman Korecky. 2022. *Ontologies in Industry 4.0: Standards, Applications, and Methodologies.* Master's thesis. Department of Information Systems and Operations, Vienna University of Economics and Business.

[18] Yongxin Liao. 2013. *Semantic Annotations for System Interoperability in a PLM Environment.* Retrieved June 23, 2022 from https://tel.archives-ouvertes.fr/tel-00904822/file/20131114_Thesis_Defence_YongxinLIAO_Version_1.7.pdf

[19] OAGI. 2022. *Open Application Group (OAGi) website.* Retrieved June 23, 2022 from https://oagi.org/

[20] POSC-CAESAR. 2022. *POSC Caesar Association.* Retrieved June 23, 2022 from https://www.posccaesar.org/

[21] DNV Recommended Practice. 2021. *Assurance of sensor systems.* Retrieved June 23, 2022 from https://rules.dnv.com/servicedocuments/dnv/#!/industry

[22] Roseanne Price and Graeme Shanks. 2005. A Semiotic Information Quality Framework: Development and Comparative Analysis. In *Journal of Information Technology*, Vol. 20.

[23] READI. 2022. *Requirement Asset Digital Lifecycle Information (READI) website.* Retrieved June 23, 2022 from https://readi-jip.org/

[24] Anna-Katrina Shedletsky. 2019. *Manufacturing Wastes 10% Of The GWP Every Year. Here's Why.* Retrieved June 23, 2022 from https://www.forbes.com/sites/annashedletsky/2019/10/18/manufacturing-wastes-10-of-the-global-gdp-every-year-heres-why/?sh=5c5b80351098

[25] Swagger. 2022. *OpenAPI Specification.* Retrieved June 23, 2022 from https://swagger.io/specification/

[26] Christopher Tate. 2014. *Understanding cutting equations.* Retrieved June 23, 2022 from https://www.ctemag.com/news/articles/understanding-cutting-equations

[27] Tiangolo. 2022. *FastAPI framework, high performance, easy to learn, fast to code, ready for production.* Retrieved June 23, 2022 from https://fastapi.tiangolo.com/

[28] WAYKEN. 2022. *Feed Rate And Cutting Speed: What's The Difference In CNC Machining.* Retrieved June 23, 2022 from https://waykenrm.com/blogs/feed-rate-and-cutting-speed-difference-in-cnc-machining/