

Linked Data Architecture for Assistance and Traceability in Smart Manufacturing

Marko Friedemann^{1,*}, Ken Wenzel^{1,**}, and Adrian Singer^{1,***}

¹Fraunhofer-Institute for Machine Tools and Forming Technology IWU, Reichenhainer Straße 88, 09126 Chemnitz, Germany

Abstract. Traceability systems and digital assistance solutions are becoming increasingly vital parts of modern manufacturing environments. They help tracking quality-related information throughout the production process and support workers and maintenance personnel to cope with the increasing complexity of manufacturing technologies. In order to support these use cases, the integration of information from different data sources is required to create the necessary insights into processes, equipment and production quality.

Common challenges for such integration scenarios are the various data formats, encodings and software interfaces that are involved in the acquisition, transmission, management and retrieval of relevant product and process data.

This paper proposes a Linked Data based system architecture for modular and decoupled assistance software. Its web-oriented approach allows to connect two usually disparate data sets: semantic descriptions of complex production systems on the one hand and high-volume and high-velocity production data on the other hand. The proposed concept is illustrated with a typical example from the manufacturing domain. The described End-of-Line quality assessment on forming machines is used for traceability and product monitoring.

1 Introduction

Today's production environments and the Internet of Things provide little standardization beyond the range of machine-to-machine (M2M) protocols. Interoperability for the digital transformation of the manufacturing domain is the focus of the Reference Architecture Model I4.0 (RAMI4.0) [1] and the Industrial Internet Reference Architecture (IIRA) [2]. Both define architectural concepts, patterns and frameworks for information and communication technologies in the context of Industry 4.0 (I4.0) or the Industrial Internet of Things (IIoT). Unfortunately, while they refer to communication interface standards like OPC-UA, they do not propose or define standards for data representation or storage formats.

Machines and production systems are therefore often using proprietary solutions for data storage and operators of large and heterogeneous production lines are confronted with different systems, databases or files in varying formats. These are largely purpose-built for specific access patterns and interfaces and their lack of flexibility causes high effort for integration scenarios like the development of traceability systems or industrial assistance solutions. This

*e-mail: marko.friedemann@iwu.fraunhofer.de

**e-mail: ken.wenzel@iwu.fraunhofer.de

***e-mail: adrian.singer@iwu.fraunhofer.de

paper proposes a data architecture based on the concepts of Linked Data [3, 4] as a basis for modular and independent assistance software and traceability solutions. Where applicable, process data is represented as time series using Linked Data concepts while semantics of production data and descriptive information are represented using lightweight ontologies. This facilitates the combination of data from multiple sources and enables the deduction of new information. The proposed concept is illustrated with a typical example from the manufacturing domain, an End-of-Line quality assessment system on forming machines.

2 Related Work

Fundamental work for the design of an ontology to describe sensors and observations was carried out within the W3C Semantic Sensor Network Incubator group (the SSN-XG) [5]. The developed SSNX ontology can be used to describe sensors with their capabilities, the methods used for sensing as well as the related observations. As a joint work of W3C and OGC the ontology was later improved and split into a lightweight core module, the so-called SOSA (Sensor, Observation, Sampler, and Actuator) ontology, and an extension ontology called SSN (Semantic Sensor Networks) [6]. The SOSA+SSN ontologies assume that both the structure of the system under observation as well as the measurement data itself are expressed in RDF.

The works [7] and [8] introduce a concept to store structural data about the system under consideration and time series data in different databases. The structural data is represented and stored as RDF while time series data is stored in purpose-built databases. In this regard, these works are similar to our approach but they do not define clear semantics for transforming the time series data to and from an RDF representation and they propose custom query engines to access the combined store.

A generic approach for representing time series data via Representational State Transfer (REST) principles and RDF on the web is described in [9]. The method uses Linked Data principles to publish raw and aggregated measurement data from microgrids along with metadata for physical units and time ranges under consideration. The developed ontology is called SEAS-eval and models the GECAD microgrid in the Institute of Engineering - Polytechnic of Porto (ISEP/IPP).

In the domain of building information modeling (BIM) the Brick metadata schema [10] as well as the Building Topology Ontology (BOT)¹ represent two complementary approaches for modeling the structure of buildings together with contained sensor and actuator systems. Both use RDF and OWL for the representation of descriptions. While BOT as a minimal ontology mainly focuses on hierarchical and topological structure of buildings the Brick ontology does also define vocabulary to describe electrical devices like lighting, sensors as well as control relationships within a building. BOT can be seen as some sort of upper ontology for BIM that can be combined with ontologies like SOSA or Brick to describe more aspects within a certain domain. Neither Brick nor BOT define vocabulary for the representation of measurements.

3 Data Integration for Traceability and Assistance

The acquisition and consumption of data is the basic common denominator for I4.0/IIoT solutions, condition monitoring, assistance and traceability systems alike. They differ in the scope of the data they require, and as outlined above, often use purpose-built representations and storage concepts. Nevertheless, since the different pools of data are related (equipment wear

¹<https://w3c-lbd-cg.github.io/bot/>, visited: 10/05/2019

might increase friction and required forces, cause higher power usage and ultimately influences product quality etc.), it is becoming increasingly important to find appropriate ways to share the data along the manufacturing processes. This is especially important for traceability solutions and assistance systems that consider the influences of individual process steps on finished parts or products. Hidden quality problems of parts are often only uncovered at later stages of the manufacturing chain, e.g. during the assembly process or at the paint shop. To identify possible causes for quality issues, process data of preceding manufacturing steps needs to be investigated. For gaining deep insights into those, aggregated measurements are usually insufficient. Therefore, to characterize the development of process variables during a process step, sequences of measurements need to be collected and stored, often with high sample rates and over long durations.

A method for RDF-based modeling of high-level traceability data regarding sequences of process steps with their raw materials, machines and workers for manufacturing a specific part is described in [11]. While the method is able to model and query the individual activities, it does not provide any means for representing more detailed process data like the development of pressing force for a single stroke.

Throughout the rest of the paper, we will use a simple example that - for reasons of brevity - omits the provenance concepts of OntoPedigree. The reduced structural RDF model just represents a single column equipped with a strain gauge sensor for measuring forces, with an example query to retrieve all sensors, shown in figure 1.

Additionally, figure 2 shows three strokes worth of actual measurements from the actual press, visualized in a Grafana dashboard. It contains data for various forces, positions and other properties, and wed like to leverage the RDF structural model to select from and filter the measurement data.

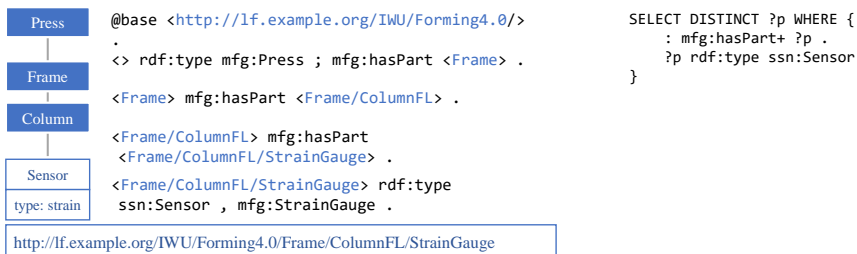


Figure 1: Example Query to receive all sensors

4 Linked Time Series Data

When using SSN or rather the more lightweight SOSA ontology to represent measurements, each measurement requires at least 5 triples as shown in Listing 1.

Especially with high sample rates, inserting the corresponding amount of triples can become an issue due to updating the index structures used by the RDF triple stores, which are optimized for graph storage. The RDF specification states that

”The RDF data model is atemporal:
 RDF graphs are static snapshots of information.“ [12]

Although this does not mean that RDF is cannot be used to represent temporal information, as supported by SOSA/SSN for example, it indicates that RDF has no native support for storing temporal data.

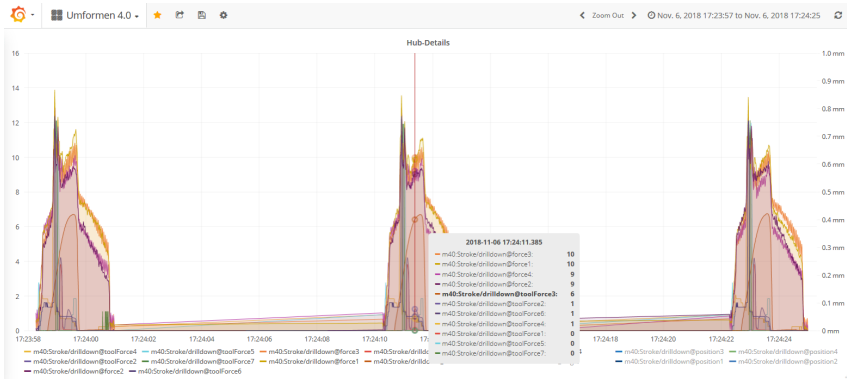


Figure 2: Example Query to receive all sensors

```
<observation/12> a sosa:Observation ;  
  sosa:hasFeatureOfInterest <oven> ;  
  sosa:observedProperty <temperature> ;  
  sosa:resultTime "2019-10-01T09:57:00.230"^^xsd:dateTime ;  
  sosa:hasSimpleResult "230.7"^^xsd:double .
```

Listing 1: Example of a SOSA encoded observation

Apart from inserting the triples, data retention can also become an issue in use cases with high volume and long duration, or in scenarios with tiered storage requirements where data should regularly be compressed, archived or otherwise be removed from the live dataset, as might be the case with traceability solutions employing some long-term cold storage.

Specialized time series databases as InfluxDB, OpenTSDB and others are able to efficiently store large amounts of measurements but suffer from only supporting limited metadata about them. While it is possible to index a collection of measurements by giving them a unique name and, as in the case of InfluxDB, a set of string-valued tags, time series databases do not support the representation of more complex metadata. The latter may include the structure of the machines or sensor systems, their physical locations, the units of measurement used, as well as calibration or reference parameters, thresholds and limits.

In summary, both technologies are optimized for one part of the overall problem and while they can be used to address the other part, the resulting solutions exhibit a number of issues that stem from bending the underlying concepts to fit the requirements. It would be beneficial to combine both technologies such that their specific strengths can be leveraged and their weaknesses be mitigated by each other.

We therefore propose to

1. consequently use Linked Data principles
2. represent products, processes and resources as Linked Data
3. describe measurement data in terms of Linked Data
4. share common terms between both representations
5. use a query language capable of joining the data

and we call the result *Linked Time Series Data*.

Each measurement is encoded as a 4-tuple (S, P, T, O) , which is conceptually identical to a SOSA Observation (cf. Listing 2), and stored in a TSDB backend.

```
<observation/23> a sosa:Observation ;
  sosa:hasFeatureOfInterest ?S ;
  sosa:observedProperty ?P ;
  sosa:resultTime ?T ;
  sosa:hasSimpleResult ?O .
```

Listing 2: Mapping of 4-Tuples to SOSA

Subject S and predicate P are expressed as URIs and reference elements from the RDF triple store. Together with the timestamp T , they form the key for the time series entries as shown in Figure 3.

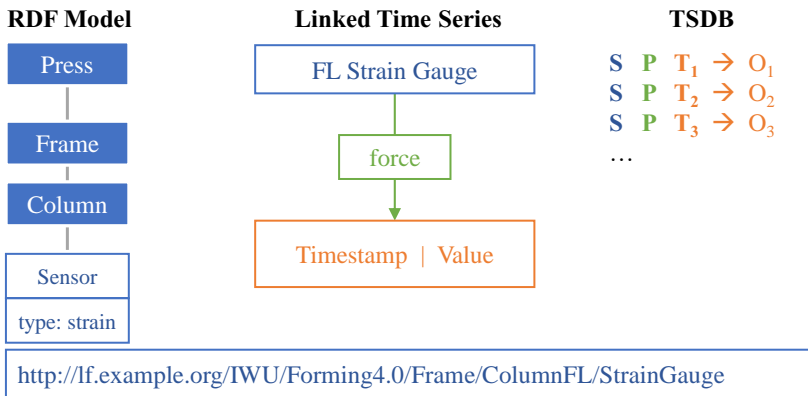


Figure 3: Linked Time Series Data Example

The shared elements S and P constitute the link between the metadata in RDF and the measurement data in the time series database. The metadata can provide descriptions of the data sources (eg. physical sensors with types and locations in a hierarchy) as well as about the observed properties (eg. physical unit). The measurement data thereby becomes self-descriptive, instead of requiring prior knowledge or explanation about mappings of opaque identifiers or labels.

The measurement data represented as 4-Tuples (S, P, T, O) naturally map to RDF as shown in Listing 3 by using the SPARQL notation for graph patterns.

```
?S ?P [ tsdb:time ?T ; tsdb:value ?O ] .
```

Listing 3: RDF representation of a simple measurement with a single value.

If the value O is not only a simple literal value but an object consisting of n property-value pairs in the form of $O = ((P_1, V_1), \dots, (P_n, V_n))$ then the RDF representation uses the properties P_1 to P_n instead of the single `tsdb:value` property.

```
| ?S ?P [ tsdb:time ?T ; ?P1 ?V1 ; ... ; ?Pn ?Vn ] .
```

Listing 4: RDF representation of a complex measurement with multiple values.

5 Queries

While the capability to store measurement data and to tag or annotate it with structural and other metadata is an essential part of our solution, capabilities to query for and retrieve the data are even more important. This is because the data is acquired and then written to the storage only once, where it becomes accessible to a number of different use cases and can therefore be retrieved many times in different contexts, matched by various criteria or applied as further selection criteria itself.

For accessing the *Linked Time Series Data* as RDF, we are using SPARQL [13] as query language. A SPARQL endpoint in front of a time series database allows to query and filter individual data points for things (cf. *sosa:hasFeatureOfInterest*) and their observed properties (cf. *sosa:observedProperty*) as depicted by Listing 5.

```
| SELECT ?time ?value WHERE {  
  <StrainGaugeFL> mfg:force ?o .  
  ?o tsdb:time ?time ; tsdb:value ?value .  
} LIMIT 100
```

Listing 5: Query data points with SPARQL.

In combination with a feature called Federated Queries, that allows a SPARQL processor to run distributed queries against multiple SPARQL endpoints, we are able to combine a query against the *structural model* of a system with another query that retrieves related *measurement data* from a time series database. An example is given by Listing 6 where the sensors contained within <FormingPress> and related values for the observed property *mfg:force* are retrieved.

```
| SELECT ?p ?time ?value WHERE {  
  <FormingPress> mfg:hasPart+ ?p .  
  ?p a mfg:Sensor .  
  SERVICE tsdb:sparql {  
    ?p mfg:force ?o .  
    ?o tsdb:time ?time ; tsdb:value ?value .  
  }  
} LIMIT 100
```

Listing 6: Combined query for model and related measurements.

6 Performance Considerations

We have applied the proposed solution to a dataset from a QA & assistance system (End-Of-Line testing) containing about 230,000,000 data points collected for about 5,200,000 parts manufactured throughout a 2-year period. It consists of optical measurements (size, location) and their deviations for key product features and a (relatively small) number of related calibration parameters and validation thresholds (< 1000 different entries). The legacy system

is using a relational database for data storage and retrieval, and requires multiple expensive SQL JOIN operations for the visualization of quality trends, flagging of defect parts and the explorative analysis towards the possible causes.

```
SELECT ?time ?flag ?ident ?pref ?pos ?nn ?vt WHERE {
  ?preset a lf:preset ; lf:source ?src ; lf:id ?pref ;
  lf:pos ?pos ; lf:nest ?nn ; lf:vtype ?vt .

SERVICE tsdb:sparql {
  VALUES ?flag { 0 }
  ?src lf:measurement [
    tsdb:time ?time ;
    lf:flag ?flag ;
    lf:ident ?ident ;
    lf:pref ?pref
  ] .
}
} LIMIT 25
```

Listing 7: Query QA data with SPARQL.

Using our Linked Time Series Data approach, we were able to reduce the physical size of the dataset on disk to about 30% compared to the SQL storage size, while achieving similar query execution times for a number of typical patterns. It is intended to connect the dataset with other data sources for process data, to apply machine learning to reduce or eliminate the need for manual exploration of causes for faults or problems, and to support faster decisions for adjustments along the process chain to increase the overall yield.

7 Conclusion

This paper proposes a new method to combine and query complex metadata and temporal data using RDF and TSDB called *Linked Time Series Data*. In contrast to the related work this approach is not restricted to any domain like energy data or BIM. By using SPARQL as query language the retrieval and analysis of temporal data can incorporate complex filter criteria, for example structural information about a production system, while keeping the benefits of time series databases for temporal data. Tests with a real-world dataset have shown a lowered memory usage in comparison to a standard SQL database while RDF allows the flexible representation of arbitrary metadata.

The natural mapping between RDF and the representation within a time series database allows to formulate easily understandable queries that can incorporate knowledge from RDF-based models of production systems, processes and products. Thus, this combination enables the creation of generic model-based assistance tools.

While the linking of RDF and time series data is possible with our approach, it is limited in regard to possible retrieval paths. Currently it is only possible to efficiently retrieve time series data by subject, predicate and time. Our future work will investigate how specific types of time series data, for example events, could be partially indexed by certain properties while keeping full compatibility with RDF. This would allow to efficiently execute queries like "Retrieve all *errors* from the machine log within the last week."

Further work will also investigate a formal alignment with SOSA/SSN to allow the representation of metadata according to those ontologies and the possibility to transparently store SOSA observations within time series databases.

Acknowledgements

This work was supported by the Fraunhofer High Performance Center Smart Production and Materials and the research project RESPOND which is funded within the program "IKT 2020 Forschung für Innovationen" by the German Federal Ministry for Education and Research (BMBF).

References

- [1] M. Hankel, B. Rexroth, *The reference architectural model industrie 4.0 (RAMI 4.0)*, ZVEI, April (2015)
- [2] S.W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, G. Bleakley et al., *Industrial internet reference architecture*, Industrial Internet Consortium (IIC), Tech. Rep (2015)
- [3] T. Berners-Lee, *Linked data – design issues* (2006), <http://www.w3.org/DesignIssues/LinkedData.html>
- [4] C. Bizer, T. Heath, T. Berners-Lee, *Linked data-the story so far*, International journal on semantic web and information systems **5**, 1 (2009)
- [5] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog et al., *The SSN ontology of the W3c semantic sensor network incubator group*, Journal of Web Semantics **17**, 25 (2012)
- [6] A. Haller, K. Janowicz, S.J.D. Cox, M. Lefrançois, K.L. Taylor, D.L. Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, C. Stadler, *The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation*, Semantic Web **10**, 9 (2018)
- [7] C.E. Kaed, V. Danilchenko, F. Delpech, J. Brodeur, A. Radisson, *Linking an Asset and a Domain Specific Ontology for a Simple Asset TimeSeries Application*, 2018 IEEE International Conference on Big Data (Big Data) pp. 4182–4188 (2018)
- [8] C.E. Kaed, M. Boujonner, *FORTE: A Federated Ontology and Timeseries Query Engine*, 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) pp. 983–990 (2017)
- [9] L. Gomes, M. Lefrançois, P. Faria, Z. Vale, *Publishing real-time microgrid consumption data on the web of Linked Data*, in *2016 Clemson University Power Systems Conference (PSC)* (2016), pp. 1–8
- [10] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal et al., *Brick: Metadata schema for portable smart building applications*, Applied Energy **226**, 1273 (2018)
- [11] M. Solanki, C. Brewster, *OntoPedigree: Modelling pedigrees for traceability in supply chains*, Semantic Web **7**, 483 (2016)
- [12] R. Cyganiak, D. Wood, M. Lanthaler, *Rdf 1.1 concepts and abstract syntax*, W3C Recommendation (2014)
- [13] S. Harris, A. Seaborne, E. Prud'hommeaux, *Sparql 1.1 query language*, W3C Recommendation (2013)