
Systematic design and implementation of a semantic assistance system for aero-engine design and manufacturing

Sonika Gogineni*, Jörg Brünnhäußer,
Kai Lindow and Erik Paul Konietzko

Department of Virtual Product Creation,
Fraunhofer Institute for Production Systems and Design Technology,
Berlin, Germany

Email: Sonika.Gogineni@ipk.fraunhofer.de

Email: Joerg.Bruennhaeusser@ipk.fraunhofer.de

Email: Kai.Lindow@ipk.fraunhofer.de

Email: ErikPaul.Konietzko@ipk.fraunhofer.de

*Corresponding author

Rainer Stark

Chair Industrial Information Technology,
Technische Universität Berlin,

Berlin, Germany

Email: Rainer.Stark@ipk.fraunhofer.de

Email: Rainer.Stark@tu-berlin.de

Jonas Nickel and Heiko Witte

Rolls-Royce Deutschland,
Eschenweg 11, 15827 Blankenfelde-Mahlow,
Dahlewitz, Brandenburg, Germany

Email: Jonas.Nickel@rolls-royce.com

Email: Heiko.Witte@rolls-royce.com

Abstract: Data in organisations is often spread across various Information and Communication Technology (ICT) systems, leading to redundancies, lack of overview and time wasted searching for information while carrying out daily activities. This paper focuses on addressing these problems for an aerospace company by using semantic technologies to design and develop an assistance system using existing infrastructure. In the aero-engine industry, complex data systems for design, configuration, manufacturing and service data are common. Additionally, unstructured data and information from numerous sources become available during the product's life cycle. In this paper, a systematic approach is followed to design a system, which integrates data silos by using a common ontology. This paper highlights the problems being addressed, the approach selected to develop the system, along with the implementation of two use cases to support user activities in an aerospace company.

Keywords: knowledge management; assistance system; semantic integration; machine learning; ontologies; industrial implementation; manufacturing; product data management; heterogeneous data; interoperability.

Reference to this paper should be made as follows: Gogineni, S., Brünnhäußer, J., Lindow, K., Konietzko, E.P., Stark, R., Nickel, J. and Witte, H. (2021) 'Systematic design and implementation of a semantic assistance system for aero-engine design and manufacturing', *Int. J. Metadata Semantics and Ontologies*, Vol. 15, No. 2, pp.87–103.

Biographical notes: Sonika Gogineni is a Research Assistant at Fraunhofer Institute for Production Systems and Design Technology, Berlin. She has a master's degree in production engineering. Her research includes topics such as smart products, smart services and semantic contextualisation. She is currently pursuing her PhD in these topics.

Jörg Brünnhäuser is a Research Assistant at Fraunhofer Institute for Production Systems and Design Technology, Berlin.

Kai Lindow is Head of the Information and Process Control department at Fraunhofer Institute for Production Systems and Design Technology, Berlin, Germany.

Erik Paul Konietzko is a student assistant at Fraunhofer Institute for Production Systems and Design Technology, Berlin.

Rainer Stark is the head of division of Virtual product creation at Fraunhofer Institute for Production Systems and Design Technology, Berlin.

Jonas Nickel is a Chief of Quality and Continuous Improvement at Rolls-Royce Deutschland.

Heiko Witte is the Business Development Manager Digital at Rolls-Royce Deutschland.

This paper is a revised and expanded version of a paper entitled 'Semantic assistance system for providing smart services and reasoning in aero-engine manufacturing' presented at 'Metadata and Semantic Research Conference', Marconi University, Rome, 28–29 October 2019.

1 Problem statement

Digitalisation is a stepwise process for most companies, thus it needs to build on existing Information Technology (IT) infrastructure and the installed Information and Communication Technology (ICT) tools and systems. These tools and systems are often not fully compatible with each other. This results in isolated solutions suitable to activities of only one group of users. To manage this diverse spread of multiple systems, approaches such as Product Lifecycle Management (PLM) and Enterprise Resource Planning (ERP) aim towards integrating them. The main advantage of connected systems is increased traceability of data across the lifecycle of the product. This in turn helps in managing products of mass customisation, achieving higher quality, reducing project failure rates, managing production and delivering efficiently (Liu et al., 2010). These benefits result in minimising overall manufacturing costs, which is an important goal of manufacturing industries. In addition, an efficient distribution of data across organisations globally can be achieved, which in turn leads to a seamless collaboration across teams (Zammit et al., 2017). A further desired benefit in the aerospace industry is the possibility to enable a much better understanding of product characteristics and their variations.

While the potential benefits are significant (Peruzzini et al., 2012), obstacles faced in practical implementation of corporate PLM systems often impede widespread use. Finding common ground in understanding the concepts and agreeing on the semantics is a major challenge for the implementation and usage of a holistic, integrated data landscape including PLM and ERP. In addition, customisation of system landscape and interfaces for individual user activities is expensive. However, the core goal that has to be achieved is a 'centralised single version of the truth' (Zammit et al., 2017, p.63). In this paper,

we focus on three factors, which pose a challenge to the implementation of such connected systems, which is also visualised in Figure 1 (Gogineni et al., 2019).

- 1) *Business environment diversity*: PLM and ERP systems are usually designed around a static, monolithic company and business process model, furthermore these systems are optimised for data consistency. However, companies are facing a continual evolution of their business environment: other companies are acquired and integrated, new customer and regulatory requirements emerge, the business model evolves, etc. These evolutions require company processes and their supporting IT systems to adapt quickly. Often, the required timelines for implementation of these changes exceed the change management capability of existing enterprise systems. As a result, current company IT ecosystems often consist of various systems holding auxiliary data in addition to the key enterprise systems. This spread of data leads to difficulties in searching for the right and up-to date information. This is not just tedious for users, but also demotivating and suboptimal. The spread of data also hampers the reusability of data. Within Rolls-Royce additional data and information are captured outside the large enterprise systems by step-wise digitisation of systems and processes, as well as by the introduction of modern, digital work concepts like Scrum. Bringing data from various systems together and creating a useful information for the engineers is a complex endeavour that requires support by means of an intelligent and novel approach.
- 2) *Heterogeneity of data*: Existing information is often available in various formats depending on the tool which manages the data. The formats might be proprietary

formats, adding to the difficulty of interoperability. Hence, to achieve connectivity across tools, it is important to work with heterogeneous data format conversion and integration (Holler et al., 2016; Zammit et al., 2017). Such integration not only enables data usage across the product lifecycle but also helps in linking product development information with other functional groups. Within aero-engine design and manufacturing, the functional groups need to be highly specialised due to the complexity of the product. However, the focus on the specialist's issues sometimes prevents cross-functional thinking when using available data and information. If those are of different formats the cross-functional collaboration becomes even harder. Therefore, overcoming the heterogeneity of the data is becoming a key enabler for the necessary cross-functional design, manufacturing and systems engineering within an aerospace company.

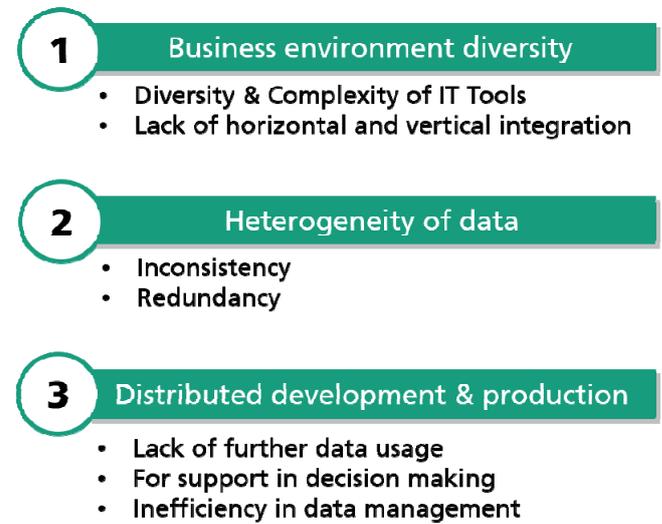
The cross-functional and cross-site design and manufacturing give rise to the third challenge of

- 3) *Distributed development and production*: On data level, it requires a connected and up-to date industrial environment and, in addition, an optimised usage of digitalised data artefacts. Data are important assets that support decision making. From a user point of view, a wide variety of individuals with different backgrounds, expertise, views, perspectives and goals come together. This is where a collaborative and customised groundwork needs to be established. From an organisational point of view, various drivers that are often opposed to each other intermingle (e.g. market requirements, customer needs, design specifications, production restrictions and quality requirements). These drivers need to be understood across all organisations involved, interlinked and managed on a common defined data level. Although PLM systems are constantly expanding their range of functions or models, they fall behind when it comes to providing user-specific or customisable functionality in a cross-functional design, manufacturing, purchasing or service context (Peruzzini et al., 2012). Within Rolls-Royce, those requirements and drivers will need to be understood as a whole within an increasingly competitive environment in order to achieve more sustainable product and service attributes within the aerospace industry.

Summarising the activities that have taken place in the project Cockpit 4.0 (EFRE funded), this paper describes the goal and approaches chosen for this project, whose aim was to achieve the advantages of integrated PLM. These advantages should be realised by small-scale implementations rather than large-scale changes and implementation efforts. Nevertheless, the benefits also needed to be scalable through customisation to fulfil user needs and their activities. This combination for challenges led to the research interest in developing a modular architecture based on semantic technologies that can be directly integrated

into existing company infrastructures. Hence, this paper includes a detailed methodological development of the modular architecture and the associated assistance system.

Figure 1 Challenges of digitalisation in manufacturing industries



This paper is structured such that Section 2 presents the current activities and research addressing the challenges, followed by Section 3, presenting the scientific approach chosen for further research and development of the solutions. Section 4 handles the practical implementation of collecting requirements and generating use cases. Section 5 illustrates the implemented architecture and focuses on the development of the prototypical solution. Section 6 presents the Machine Learning (ML) applications and preliminary results. The implementation of the demonstrator is detailed in Section 7. The evaluation of the prototype/demonstrator is detailed in Section 8. Section 9 describes findings. Finally, Section 10 summarises the essential aspects of the paper and provides outlook for the future and conclusions.

2 State-of-the-art

Different paradigms have emerged for next generation manufacturing systems, considering the heterogeneity of data from concepts such as smart manufacturing, cyber-physical production systems, Industry 4.0 (I40) or cloud manufacturing (Moghaddam et al., 2018; Zeid et al., 2019). In addition, architectural concepts such as the Reference Architecture Model for Industry 4.0 (RAMI4.0), Industrial Internet Reference Architecture (IIRA) or National Institute of Standards and Technology (NIST) are growing rapidly. These are overall architectures for interoperable manufacturing divided in different layers, views or classifications (depending on the different components used, functions or processes) and descriptions of their interference (Grangel-González, 2019; Moghaddam et al., 2018; Zeid et al., 2019). Although the

architectures provide a good starting point to consider the important elements, they are abstract and lack in providing detailed implementation recommendations, especially on existing architectures of enterprises.

Out of the manifold possibilities to handle and integrate heterogeneous data, we concentrate on ontology-based applications, since ontologies allow to structure semantics in data. They are a popular solution for cross-enterprise and – application information modelling, to integrate heterogeneous data from different domain specific sources (Cruz and Xiao, 2005). They define concepts as simplified abstract view of the world, their mutual relations and provide basis for logical reasoning through axioms or rules (Mizoguchi, 2003). An ontology can be described as ‘explicit specification of a shared conceptualisation’ (Staab and Studer, 2009; Studer et al., 1998).

In relation to PLM concepts, capturing Product Lifecycle (PLC) data is a key element. There have been several high-level PLC ontologies achieved by mapping and integration of domain specific ontologies (e.g. Bruno et al., 2015, 2016; Otte et al., 2019; Usman et al., 2011). However, in the Cockpit 4.0 project we wanted to follow domain based prototypical implementation, hence we follow the bottom-up approach. This is further discussed in Section 3. A PLC approach would build on the results of this and would thus at best be an extension of existing research questions.

There have been different approaches to develop domain ontologies such as those (Mountantonakis and Tzitzikas, 2019). focusing on a certain main aspect in the process (e.g. data set type, functions to provide, integration substance, integration or other auxiliary services) and others (Grangel-González, 2019) which survey on the suitability for standards and standardisation. Concerns highlighted for the majority of applications are the scalability of the investigated approaches (Mountantonakis and Tzitzikas, 2019) and lack of specificity to individual problems/use cases of different domains, reasoners or rule engines for transformation, validation and domain knowledge discovery (Grangel-González, 2019).

For developing individualised use case-based ontologies, different general approaches in architectural design for Ontology-Based Data Integration (OBDI) can be summarised as follows: (Ekaputra et al., 2017; Wache et al., 2002). In a *single ontology approach*, all data sources are integrated into one global ontology. A *multiple ontology approach* integrates each data source into a separate local ontology, which will be aligned to each other using semantic mappings. A *hybrid approach* also considers different local ontologies, although they will not be aligned independently, but share a common

domain vocabulary. The *Global as View* approach is rather similar to this, but the local ontologies will be integrated into a global ontology transformed with rules. As the project concentrated on a focused domain-based implementation, the single ontology approach was chosen, which also has the flexibility to be integrated into a global ontology later on.

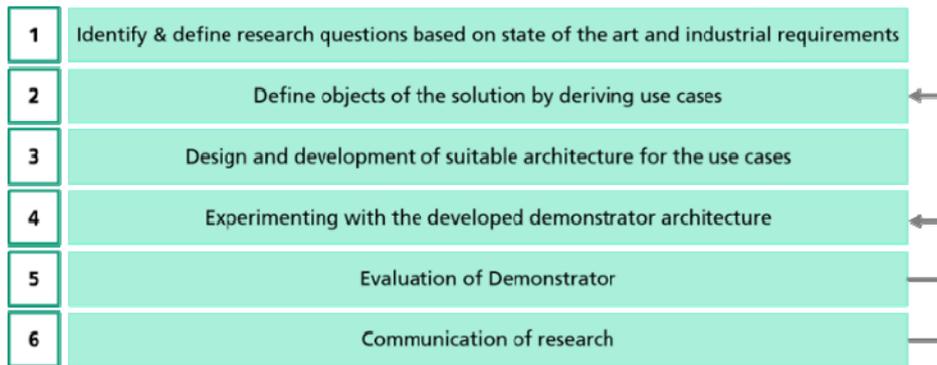
To access and handle huge amounts of data, Ontology-Based Data Access (OBDA) is a paradigm growing in prominence. OBDA maps data sources into semantically structured architectures, (Ekaputra et al., 2017; Kharlamov et al., 2015). Reasoning is carried out on terminological level and not on the data itself, which is why neither incompleteness nor scaling of data poses problems in this approach. Likewise, the data is only mapped in virtualised form. This offers advantages concerning storage and the possibility to manage and maintain the original individual sources separately while also providing the user with a global up to date and unified vocabulary at the same time (Pan et al., 2017).

There have been several attempts at industrial implementation of ontologies (Adams et al., 2000; Ast et al., 2014; Handler, 2010), which demonstrate the main advantage of connecting existing data with meaningful relationships. However, the cited authors highlight the importance of having a common understanding of the individual domains, integrating the domains of interest into the global perspective, including domain experts in the development process and working with ontology engineers to develop the concept.

The necessity of such implementations to be closer in context to the end user is gaining importance (Alegre et al., 2016). Providing information in context to the activities carried out and the relevance to the problem being handled are a part of the concept contextualisation in the field of IT. There is some initial research in this field but fewer evaluated implementations (Erkoyuncu et al., 2017; Roy et al., 2016).

3 Research approach

As mentioned in Section 2, to develop a domain-based implementation of a single ontology from heterogeneous data sources, to integrate the domain experts from the start and to encourage contextualisation a systematic development approach was followed. Hence, for the prototypical implementation the Design Science Research Methodology (DSRM) for information systems was selected. This method was deemed suitable as it encompasses the applied research nature of the project. In addition, DSRM provides a methodological guideline to design, develop and evaluate the system (Peffer et al., 2007).

Figure 2 Research approach steps (Gogineni et al., 2019)

Based on the six steps of DSRM, the approach followed in the project is shown in Figure 2 (Gogineni et al., 2019). It is also important to notice that the process was highly agile and iterative, as indicated by the grey arrows in Figure 2. Evaluations and interviews with the users and the Rolls-Royce Deutschland team provided feedback for adjustments of the design to fulfil requirements. Development was coordinated with the help of an agile project management tool. Considering the end user requirements for continuous improvement, by familiarising and involving the user with the new prototype/demonstrator is one of the advantages of this development approach, which is especially beneficial in terms of change management principles.

The main Research Questions (RQ) of the project are (Gogineni et al., 2019):

- *RQ1*: How to integrate heterogeneous data sources from ICT?
- *RQ2*: How to use intelligence in terms of algorithms to provide value added smart services along industrial processes?
- *RQ3*: How can relevant context-sensitive knowledge be generated and presented to improve processes?

4 Requirements collection and generation of use cases

The first step in the development consisted of a set of workshops whose purpose was to understand the needs for the new system from different user perspectives. The workshops were held with 25 experts from various departments in Rolls-Royce Deutschland. The departments included design, production planning, supplier quality, production and service. To translate the requirements for development, 79 raw user stories were collected in the workshops. These were further grouped and prioritised. This led to the development of five generalised user stories which were subsequently developed

into five detailed use cases of interest for the project. The complete process of generation of use cases is illustrated in Figure 3 (Gogineni et al., 2019). The use cases were further detailed with interviews of relevant users. As shown in the bottom second half right of Figure 3, the interviews were guided using this template.

In the project, two use cases were chosen for the development of the prototypical system. This decision was based on the availability of data, prioritisation based on impact generated and ability to evaluate. The two use cases are namely: Voice of the Fitter (VoF) case and Concessions case. Each of them is detailed in the subsections below.

4.1 VoF use case

In Rolls-Royce Deutschland, a special team was formed called the Voice of the Fitter, to manage problems faced by fitters during the assembly of the engine. The VoF process was introduced based on the recognition that regular communication routes for problems are not sufficient to meet the timing needs of the assembly line. This can lead to unwanted build stops and potentially delayed delivery of the product. When a fitter faces a problem (missing part, missing tools, etc.) that they cannot solve by themselves, they approach the VoF Team. This dedicated team of logistics experts take up the problem and investigate to find a solution for the issue. The team often has to search through various databases to find the root cause of the problem and then take the necessary actions. This process often has to take place quickly, as the problems on the assembly line can lead to build stops and subsequently to major losses in time and cost.

Hence, this use case was categorised into: Feedback from assembly. The defined user statement for the use case is: I am working in plant logistics and want to quickly find information needed to resolve an assembly issue. In this project however, only the VoF team was considered. In Table 1, information collected from the VoF team to further define the use case is shown.

Figure 3 Use case generation (Gogineni et al., 2019)

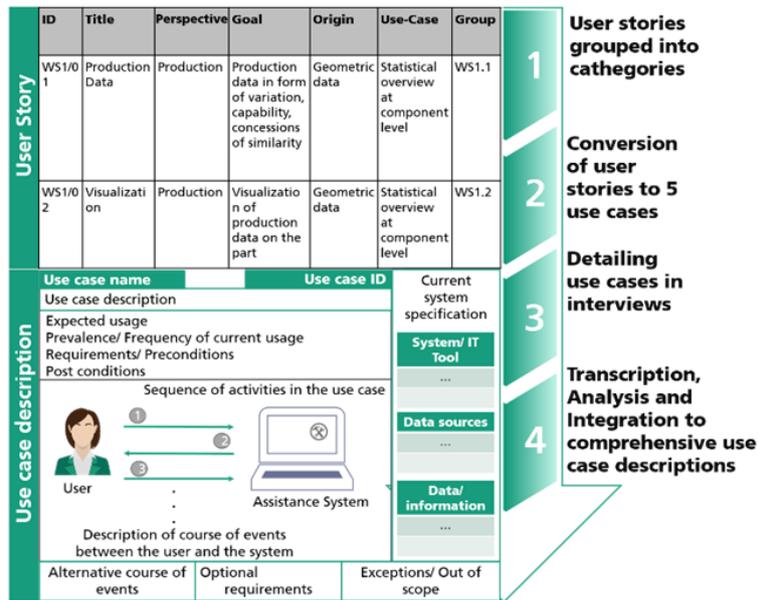


Table 1 Further information about the VoF use case

General information about the use case	
Name	Feedback from assembly
Description	I am working in plant logistics and want to quickly find information need to resolve an assembly issue.
Expected utility / benefit	Resolution of assembly issues in shorter time and with higher quality.
Frequency	Approximately 100 cases / week
Trigger	A new case is created from VoF team
Premises	All the data required to evaluate are present in digital format and on accessible systems.
Role	VoF – Team

4.2 Concessions use case

Nonconforming parts from suppliers or from the internal supply chain must be controlled in order to avoid unintended installation in the engine. For high-value parts it is common to control the acceptance of parts with minor non-conformances under concession by a qualified authority. A special team called the concessions team works on checking these parts to assess the degree of deviation. Based on the degree of deviation the parts are categorised and further actions are recommended. Understanding the functional impact of the non-conformances and defining necessary actions means the assessment requires experience and reliable data. This requires the team to access original design information, the actual part information and the historical cases to make the right decision. In Table 2, information collected from the concessions team to further define the use case is shown.

Table 2 Further information about the Concessions use case

General information	
Name	Concession
Description	I am a designer and want to evaluate a concession application in order to decide about the further usage of the part.
Expected utility/ benefit	Eased evaluation of concessions, access of historic evaluations of same or similar issues.
Frequency	Approximately 150 concessions / week
Trigger	A concession is raised for evaluation
Premises	All the data required to evaluate are present in digital format and on accessible systems.
Role	Designer, Producibility Lead, Non-Compliance Agency

5 Demonstrator design and development

Based on the use case descriptions it was clear that the existing data sources across various systems had to be integrated, processed and provided to the end user of the demonstrator. The prototypical implementation is referred to as a demonstrator in this paper. This led to the development of the system architecture with five layers as illustrated in Figure 4 (Gogineni et al., 2019). In addition, blocks illustrated in grey boxes indicate the key research topics which have to be addressed across these layers to develop a user friendly and effective demonstrator.

The layers are individually detailed in the subsections below, using the bottom-up approach, starting with the data layer. A microservice architecture was chosen for the

development of the backend. The reason being its flexibility for agile development of individual code blocks. This leads to the isolation of faults and enables collaborative development of the application (Nemer, 2019). The microservice architecture of the demonstrator is shown in Figure 5, where the front end communicates with a gateway. The gateway in turn links to the individual microservices developed for various functionalities. The Java based application was developed using the Spring Boot framework for the backend and Vue.js for the frontend.

Figure 4 Demonstrator architecture (Gogineni et al., 2019)

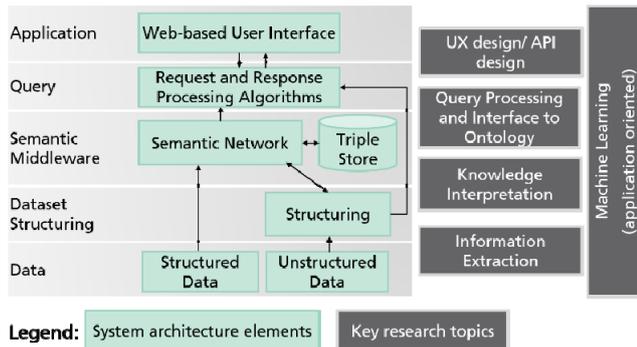
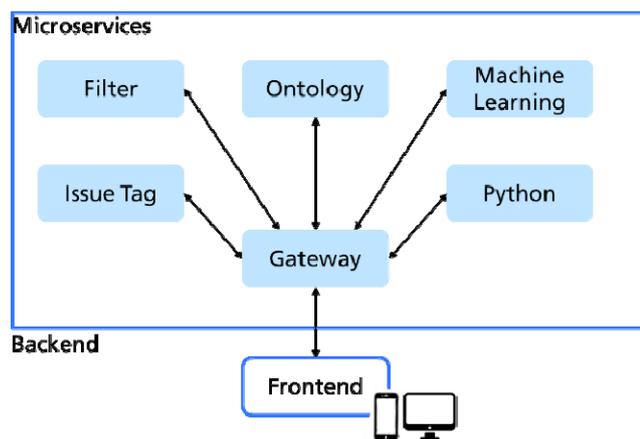


Figure 5 Complete architecture of the demonstrator



The functions the microservices carry out are further explained in the relevant layers of the demonstrator in the following subsections.

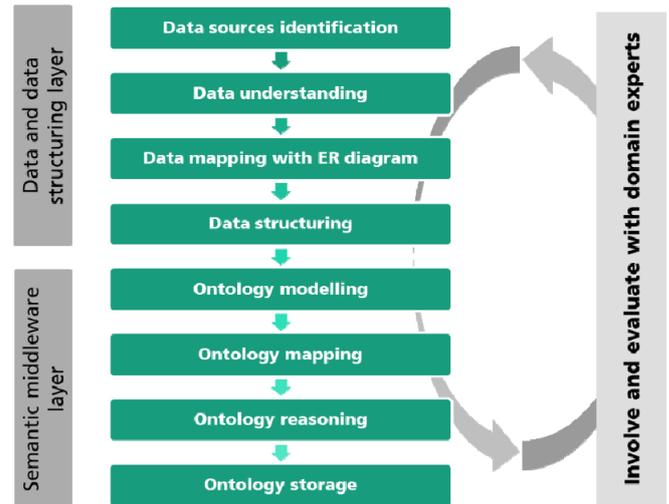
5.1 Data and data structuring layer

Over the years, the IT estate at Rolls-Royce has experienced many of the transformations and diversifications described in the introduction. In addition to the challenges faced by many industrial companies, aerospace products have extremely long life cycles – often spanning decades – during which design and configuration data need to remain available. As a result, the IT environment contains both legacy and current IT systems and applications, which leads to a heterogeneous data landscape. Furthermore, many data sources contain unstructured data such as design reports and problem resolution sheets. However, the data has to be structured to be used in a semantic context. Based on the use

cases requirements it was clear which information should be integrated into the ontology.

As the demonstrator is designed for presentation of data from various sources in context to the user, it is important to identify and understand the data which is to be presented. Figure 6 illustrates the steps followed to support the discovery of data, data structuring and the development of the semantic middleware layer. Hence, each of the elements in the figure is discussed in this subsection and in Sub-section 5.2.

Figure 6 Ontology development steps



Data source identification was already a part of capturing the use case details mentioned in Section 4. Based on interviews of the users, the information systems the users would require to access were captured. This enabled the identification of the main data sources which delivered the information required for the activities.

Data understanding was the second step, where the data sources were analysed to understand their structure, form of storage (relational type, document type, etc.) and relationships among the various data sources. To develop the ontology, main terms and concepts of the domain provided initial pointers to identify the classes of the ontology.

Data mapping is necessary in complex IT ecosystems to capture the relationship between the various data sources. An Entity-Relationship (ER) model was used, that captured actual physical entities, like parts, as well as data entities in the company databases. Owing the complexity of some of the enterprise systems, only the relevant subsets of their data models were included in the data mapping.

The model helped to clarify roles and cardinalities as well as data duplication in the source systems. One example was the attribute ‘part number’, which – even though unique on business level – was replicated as a stand-alone copy or even permitted as a free-text field. This required significant synchronisation and data consolidation work during the pre-processing.

A section of the ER diagram is shown in Figure 7. It shows three of the entities: *Part Physical*, *VoF Case* and *Concession*, and their attributes. The Primary Keys (PK) and Foreign Keys (FK) are shown in bold.

Data structuring uses the results of the data mapping to determine which tables from the existing data sources might be usable as a direct export and where additional pre-processing is necessary. This is also relevant for the architecture and the Extract-Transform-Load (ETL) strategy of any data warehouse that may be used for buffering data for the semantic middleware.

Where useful tables exist, a direct query to the source database is an option, provided a sufficient query performance can be guaranteed. Where data transformations or data cleansing is required, the data should be stored in a data warehouse, which was modelled with a MariaDB database in the case of the demonstrator. An Enterprise Service Bus was used to create an abstraction layer for the deep queries from the semantic middleware.

In the case of the demonstrator, the entity Part Physical was created to reflect serialised parts that could be used to connect to VoF cases and concessions. However, this entity used ETL extracts of the standard MARA (General Material Data) and MARC (Plant Data for Material) tables from the SAP ERP system as a basis. The Concession entity was based on the SAP QMEL (Quality Notification) table. In this case, some significant transformations were necessary in order to limit the data scope and simplify querying for the middleware.

5.2 Semantic middleware layer

This layer connects the data sources from various systems with the frontend. The semantic middleware layer is further divided into elements such as modelling of the ontology, ontology mapping, reasoning and ontology storage. These elements are illustrated and highlighted in Figure 8. In addition, the association with the layers above and below the semantic middleware layer are also illustrated. The tools used for every

element are also shown next to its respective element in the Figure 8. For example, for the modelling of the ontology, Protégé software was used. The choices for the tools selected were based on their open source compatibility, scalability, ease of integration into the selected software stack of the demonstrator and availability of guidelines and examples.

Figure 7 Snippet of ER-Model

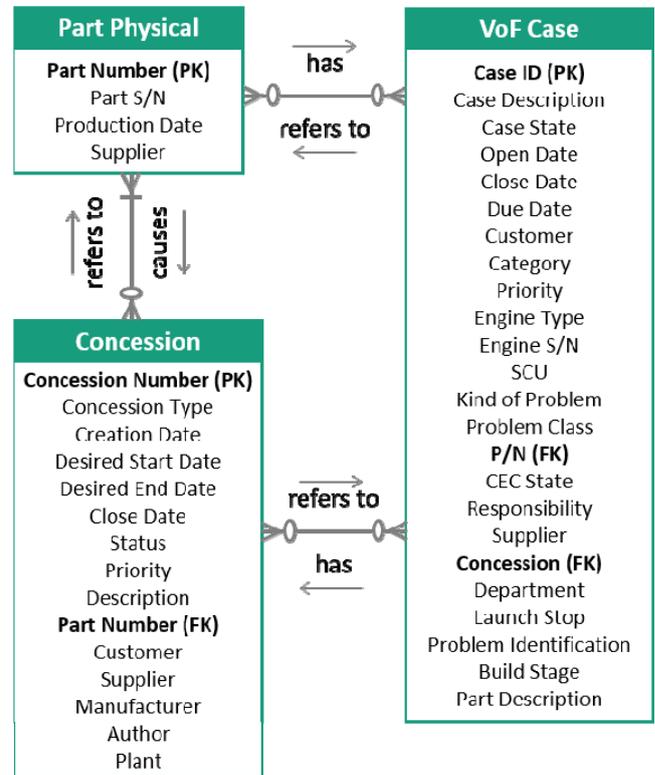
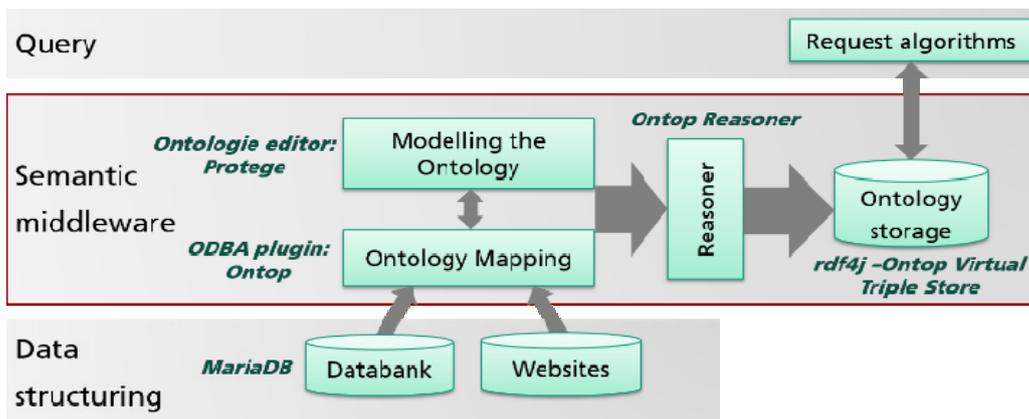


Figure 8 Semantic middleware layer architecture



Each of the elements mentioned in the semantic middleware layer are detailed in the next paragraphs.

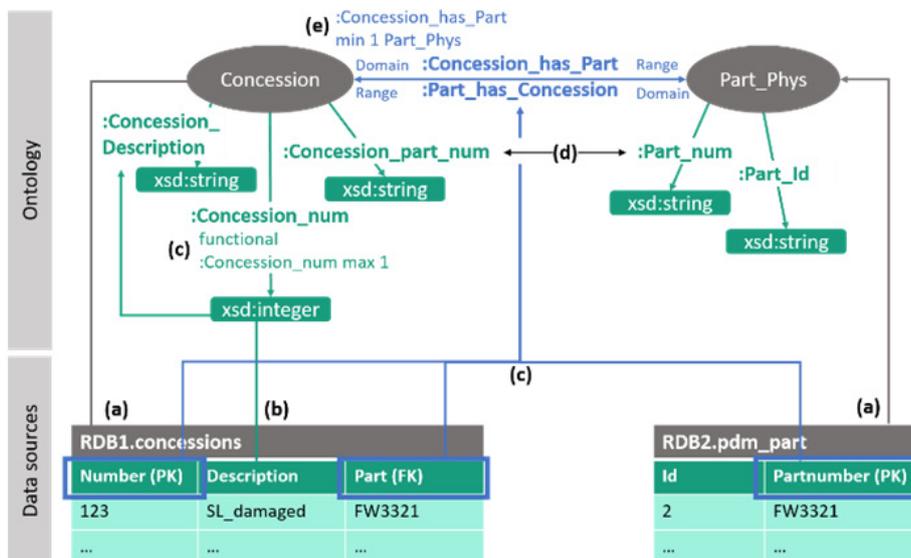
Ontology modelling was performed by following and applying transformation rules (Louhdi et al., 2013; Ren et al., 2012). The transformations enabled the identification of classes, object properties and data properties of the ontology from entities and attributes of the database. A corresponding example to each aspect of the transformation is given in Figure 9. Semantic nets are usually described in Resource Description Framework (RDF), a data model using triple patterns for knowledge description. A triple pattern thereby is built comparable to a simple sentence construction in the form: subject, predicate, object. The subject in ontological terms is also called the domain, a predicate is a property describing a relation and the object is called range of a property. Based on the type of relation, there are object properties (connecting classes to other classes) and data properties (connecting classes to datatypes) (Stuckenschmidt, 2011). An example of a triple pattern is shown in Figure 9. *Concession* :*Concession* *has* *Part* *Part_Phys*, where the *Concession* class is the domain of the object property :*Concession* *has* *Part* and the *Part_Phys* class is the range of the object property, vice-versa for the object property :*Part* *has* *Concession*.

- a) *Transforming entities*: Entities are transformed into classes or subclasses depending on if they are strong entity or weak entity. For the use cases only strong entities were transformed into classes. To ensure no overlapping, classes were disjointed. For example, in the concessions use case, the concession entity is transformed to *Concession* class in the ontology (cf. Figure 9).
- b) *Transforming basic attributes*: This applies to simple attributes that are not primary keys. The attributes provide further information about an entity (e.g. concession type attribute provides information about a concession as shown in Figure 7). In ontological terms, such attributes are represented by data properties. Its domain is the corresponding class and the range refers to its datatype in the database represented in xsd standard (XML schema

definition language, e.g. xsd:string). An attribute/entity with ‘NOT NULL’ constraint is transformed into a property/class with *min cardinality*. Unique attributes are set as *functional property* (functional means a property can only have one value). By applying these rules, for example, the attribute ‘Description’ of the concession entity is transformed into a data property, which can contain null values and is not unique (shown in Figure 9).

- c) *Transforming PK and FK*: PK and FK indicate a relation between the respective classes in the ontology. This is transformed into object properties in ontologies. Object properties have a unidirectional characteristic, while ER relations can be both unidirectional as well as bidirectional (in case of M:N relations, see Figure 7). To capture the binary characteristic the property has to be enhanced with an *InverseOf property*. As an example, the relation (object property) between *Concession* and *Part Physical* is shown in Figure 9. A PK as key attribute is set as *functional* and *max cardinality* restriction is set to 1 (as it is a unique value).
- d) *Naming convention*: Each class, data and object property has to be identifiable. Therefore, unambiguous name assignment ensures distinctiveness in transformation of ER to ontology. To ensure this, as an example for the data properties the corresponding entity was prefixed in the name.
- e) *Transforming cardinality in relations*: Referring to cardinalities on the crow’s feet in Figure 7, when represented by ‘ \leftarrow ’ symbol, then it is transformed into a *min cardinality* constraint. For example, a concession must correspond to a part. This is not always the case in Rolls-Royce, since a concession also can apply to an assembly. However, for the current use case, only the connection to physical parts is implemented requiring a *min cardinality*. Other cardinality constraints, on the other hand are represented by ‘ \leftarrow ’ symbol, meaning e.g. a part can have a concession.

Figure 9 Examples of applied transformation rules



The final modelled ontology is shown in Figure 10. Only a few data properties of the implemented ontology are shown (in green) for better readability. In addition to the classes already discussed before, a new class is introduced called *Line_item* which is associated with the concessions class. Line items are individual revisions of a particular concession and are documented using a separate tool and database.

Ontology mapping was the next step to connect the data with the ontology itself. This was done using the OBDA tool called Ontop, which is compatible with the modelling tool Protégé. As stated in Section 2, OBDA combines the advantage of Relational Databases (RDB), which is scalability, and those of ontologies, such as low maintenance and its dynamic nature (Pan et al., 2017). This is achieved by the fact that mapping avoids duplication of data and connects to the up-to-date data in the original source. A mapping

describes relation of the elements in the ontology and the original data source. This is represented in the form of a virtual RDF graph. This graph can be queried using SPARQL (SPARQL Protocol and RDF Query language), which in turn is translated into Structured Query Language (SQL) to access RDB source data (Calvanese et al., 2016). Calvanese et al. (2016) provided comprehensive insights on the functions of Ontop. An example of a mapping used in the concessions use case is shown in Figure 11. In general, the mapping contains one part describing the location of the data sources in the database (RDB) through SQL queries (cf. lower half of the ontop mapping in Figure 11). The other part defines how the data has to be associated with the ontology (cf. upper half of the ontop mapping in Figure 11). This mapping can be accessed through the SPARQL query triggered by the frontend.

Figure 10 Implemented ontology

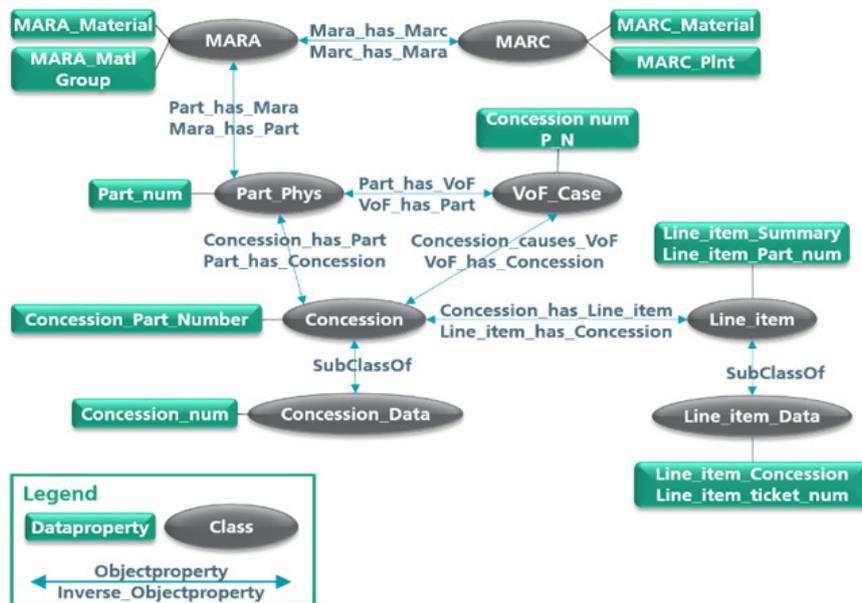
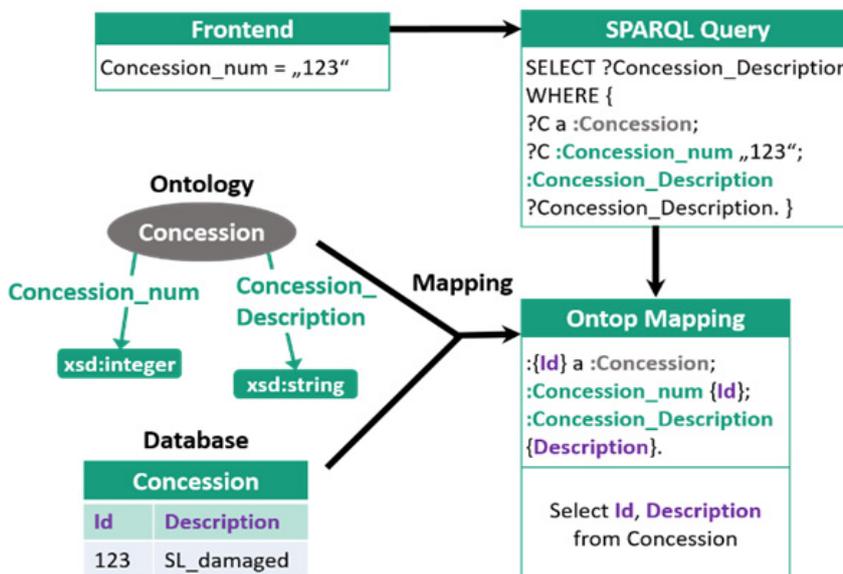


Figure 11 Mapping example



Ontology reasoning facilitates investigating if the ontology and its mapping follow the defined rules. However, there were certain challenges during the reasoning phase. As the ontology is designed using Ontology Web Language (OWL), it assumes the Open World Assumption (OWA) for drawing inferences. The assumption assumes that falsity of a statement cannot be deduced from the absence of truth alone (Staab and Studer, 2009). Therefore, a reasoner will not conclude inconsistency from missing individuals or properties, because it assumes that the needed information could be present somewhere else. This affects especially the *min cardinality* restriction. In case of missing information in the position of a PK or FK in a situation where a *min cardinality* restriction exists, the reasoner does not throw an exception error because of the OWA. This was the case for *Line_item* and *Concession* classes. To avoid the construction of inconsistent classes, meaning violating the *min cardinality* restriction, the source data was mapped into superclasses in the concession and line item case (cf. Figure 10). The superclasses (*Line_item_data* and *Concession_data*) contain all but the key data properties of respective class in the ER model needed for the relation. The domain of those key attributes was set to the actual *Line_item* and *Concession* class. Any data set mapped into the ontology that does contain all key attributes will then automatically be reasoned as entity of the actual class.

An example in the concession use case: A dataset that contains a data property *Concession_num* is not considered as an entity of the *Concession* class as long as it does not hold a value for the data property *Concession_Part_num*, since a *Concession* is restricted to have at least one *Part_Phys*. As long as this is not met, the data set will be considered as a collection of data from some incomplete concession and cannot be added to the *Concession* class at this time. An advantage of considering the incomplete data set is that, these can be identified and completed to enhance the quality of the data.

Ontology storage is facilitated through a tool called Eclipse RDF4J. In compatibility with the Ontop plugin, the type of store generated is called an Ontop virtual RDF store. The RDF4J tool enables the creation, storage, reasoning and interaction with RDF data. In addition, it acts as a SPARQL endpoint on an Apache Tomcat server, which can be accessed by the demonstrator (Eclipse RDF4J). Functions to connect and interact with the virtual store are defined in the ontology microservice of the demonstrator.

5.3 Query layer

This layer provides the frontend with requested data from the semantic middleware layer. Frontend activities trigger functions of query generation in the ontology microservice accessible over the gateway (see Figure 5). Queries generated will be sent to a SPARQL-Endpoint.

To narrow down query results, there are different filter options to choose on the frontend. Some of these options are: to select the part of interest, adjust the date range for filtering, to select state of the issue and to find cases through a text search option. Based on the selected criteria, issues are displayed on

the affected part using dots or squares. Issues here are VoF cases or concessions cases.

In addition to the actual issues, the VoF interface contains queries which look for historic cases with similar parts, problem identification or concessions. The concessions use case interface contains queries to extract detailed information about the concession from another data source (line items), historic concession cases on the same part or similarly detailed concessions. Also the queries provide information for ML algorithms for classification based on historical decisions for the user. An example of the SPARQL query is shown in Figure 12. The first part (a) provides the description and ID of historic VoF cases with the same problem identification (<ProblemID>) as the selected case. The second part (b) provides a query template for concessions with generic variables. This can be modified accordingly, based on information required by the front-end (User).

Figure 12 SPARQL-Queries for (a) VoF and (b) Concessions case

```
(a) prefix :<http://www.semanticweb.org/fraunhofer/
Cockpit4.0#>
SELECT DISTINCT
?VoF_Case_ID ?VoF_Case_Description
WHERE{
?VoF_Case a :VoF_Case;
:VoF_Problem_Identification '<ProblemID>';
:VoF_Case_ID ?VoF_Case_ID;
:VoF_Case_Description ?VoF_Case_Description.
}
(b) prefix :<http://www.semanticweb.org/fraunhofer/
Cockpit4.0#>
SELECT DISTINCT
?<Variable[1]> ?<Variable[2]>
WHERE{
?L :Line_item_Meldung '<Meldung>';
:<Dataproperty[1]> ?<Variable[1]>;
:<Dataproperty[2]> ?<Variable[2]>.
}
```

5.4 Application layer

To support a portable, accessible and flexible front end application, a web-based user interface was chosen. A web-based interface was also preferred due to the extensive number of semantic web-technologies which are already available. To design the application layer the requirements were based on: functional user requirements, user experience, design for knowledge presentation, design for interaction and visual/aesthetic requirements. The requirements were implemented as minimum viable products. This was evaluated with the end users. An agile process was followed for collecting new requirements and evaluating the implementation. This helped in achieving the optimal front end for the end user. Based on the initial requirements from the use cases the important elements of the web-interface were

established to be the: 3D-model of the part, the issue located on the model, information about the issue, context relevant information about the issue and the ability to select and filter. The web-based frontend was developed using Vue.js. The choice was made after having compared with Angular and React.js. The advantages were the compatibility with the backend tool (Spring Boot), wide spread implementation hence leading to extensive documentation of examples and relatively simpler syntax to learn. In addition, these standards enable using templates which reduces time of development and support standardised development.

6 Machine learning

The Machine Learning (ML) requires functions and co-working of several layers and is therefore applicable across the layers of the demonstrator. The following paragraphs further detail the approaches used and the results obtained through machine learning.

In order to assist the VoF team in finding a solution to an existing assembly problem, three different ML approaches have been implemented into the demonstrator. These approaches are based on the same data as the ontology. There are two relevant tables:

- The VoF cases representing the historical cases, which the VoF team has encountered and solved.
- The VoF comments which stores one to many comments for almost every VoF case.

One way of helping the VoF team is to show them similar cases to their current open one. The idea behind this is that historic cases, which are similar to the current one, could help in finding a solution. Especially the connected comments might provide hints regarding how a similar problem was solved before. To implement the first approach the data was prepared. This included merging the tables, deleting or replacing missing values, turning categorical attributes into numerical ones, scaling the data and finally using UMAP (McInnes et al., 2018) for dimension reduction.

As a next step, several clustering models were trained with different algorithms like DBSCAN, HDBSCAN (Campello et al., 2013) and k -Means. The clustering results were compared regarding their quality. This was rated with different scores such as the silhouette score proposed by Rousseeuw (1987). The best clustering generated by k -Means was chosen to be implemented into the demonstrator.

The second approach is to predict the duration of an open case. This could help the VoF team to prioritise their work – a case where a long duration is predicted could have a higher risk of causing a build stop, thus those cases should be focused on. The data were prepared in almost the same way as for the previous approach, except that a different scalar and no dimension reduction were used. Furthermore the label for the training was calculated from the start and end timestamp and the data were split into a training and a test set. After that different regression models were trained with the training data set: linear regression, lasso regression, support vector

regression (all with scikit learn (Buitinck et al., 2013)) and a model tree (Wong, 2018). For the evaluation of the trained models the test data set was used and for each model the root mean squared error was calculated. The model tree performed best and had the lowest root mean squared error.

The third machine learning approach was based on the experience that the results of some ontology queries are returning several hundreds of cases. So the goal for the last approach is to sort the results regarding their relevance, which would help the VoF team to find helpful cases faster. One way of estimating the relevance is to calculate the textual similarity between the case descriptions. If they are more similar, it is assumed that the cases are closer to each other. For this reason, three different natural language processing pipelines were set up:

- The first one uses a bag of words like Manaa and Abdulameer's (2018) approach with Jaccard (Niwanakul et al., 2013) as a distance measure
- The second one uses a self-trained word2vec model (Mikolov et al., 2013) and Word Mover's Distance (Kusner et al., 2015)
- The third one uses the word2vec model (Mikolov et al., 2013), which is pre-trained on Google News, and Word Mover's Distance

In order to estimate the most suitable approach, some results have been rated by an expert. The second, self-trained model was chosen to be the most accurate.

For the concessions use case, an ML model was trained to aid the concessions team in gathering essential information quicker. The model is used during a first assessment of the concession. In this phase the team estimates whether a diversion from a specific nominal tolerance is acceptable. Therefore, the team needs to find other concession cases that are similar to the current one and put them into context. If the actual value's deviation from the nominal value is smaller than for the other accepted cases, the concession is likely to be accepted. This is also represented in a graph to facilitate quicker decision making.

To speed up the search for relevant cases a k -nearest neighbour model was trained. During the pre-processing irrelevant data was dropped, some important missing information such as the examined nominal and actual value were extracted from a description text for each concession. The nominal and actual values that could not be extracted from the description were imputed with scikit-learn's (Buitinck et al., 2013) IterativeImputer, using single imputation with a Random Forest model with 50 trees to estimate the missing values. All data was then scaled and fed into the k -nearest neighbour model (shown in Figure 15 number 4).

For evaluation purposes, a questionnaire with two types of questions was created. In the first task an expert rated a concession for similarity to five other concessions, these had to be sorted from most similar to least similar. One of these concessions was the nearest neighbour according to the k -nearest neighbour model and the other four were chosen randomly. Two experts evaluated four examples and

they always selected the case as the most similar which was chosen by the k -nearest neighbour model as well.

In the second task the expert rated the accuracies of the nearest neighbours that were suggested by the model. Four different concessions with their five nearest neighbours were presented to the experts. Both experts considered 18 out of those 20 proposals from the k -nearest neighbour model as a good pick. Overall, the evaluation seems to prove the quality of the model.

7 Demonstrator implementation

The VoF case was implemented first, followed by the concessions case. The intention was to test the possibility of extending the demonstrator for different applications. The test was successful, the same backend architecture was used by adding the microservices relevant to concessions. The frontend was duplicated and made accessible using a login page as shown in Figure 13.

The general requirements for the frontend were the display of (as to see in Figures 14 and 15):

- 1) a 3D-model of the part
- 2) the issue located on the model
- 3) information about the issue
- 4) context relevant information about the issue
- 5) filter and selection criteria.

The frontends could also be independently coded with features relevant to the use case. This can also be seen comparing Figures 14 and 15.

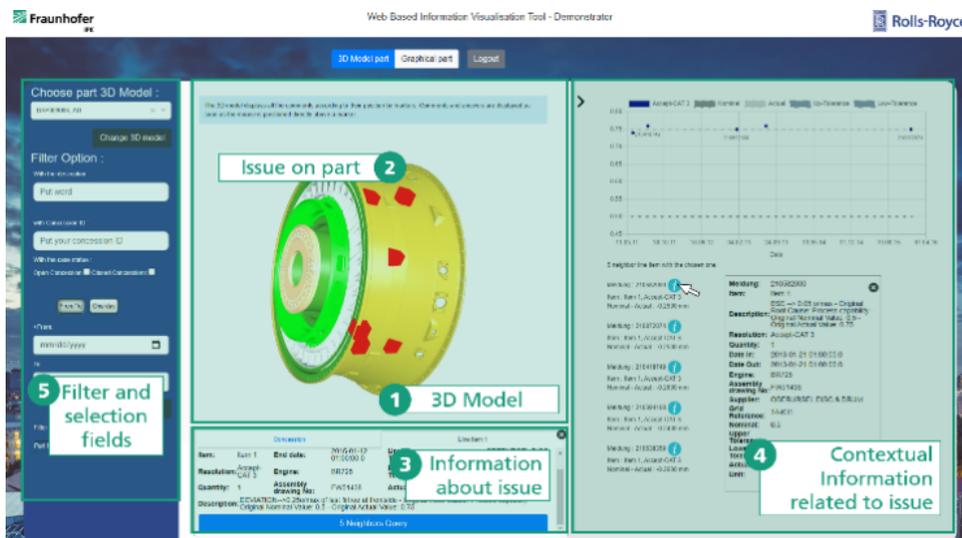
Figure 13 Login page to select the relevant use case



Figure 14 VoF case frontend



Figure 15 Concessions case frontend



The major differences between the two frontends are the options to query the ontology and the machine learning features, as they depend on the end user's needs. In the VoF case there are five query options (as blue buttons) in the box annotated with the number 3 in Figure 14. The options allow the user to access contextual information such as:

- Part number query fetches all the issues belonging to the part number under investigation. This provides the user an overview of all the problems on a particular part during assembly.
- Problem identification query finds all issues with the same problem category. This makes it possible for the user to view previous solutions or know who they can contact to solve the current issue with similar circumstances.
- Problem identification query with part number is a query similar to the previous query, but additionally filtered on a certain part number. Hence, the user can access information needed to solve an issue applicable to an explicit part.
- Concession query provides the user all VoF cases related to the same concession as the current one. If a *Part Physical* has a concession, the concession itself can hold information about problem solving.
- Cluster query provides the user with similar issues to the VoF issue under investigation, which is based on the machine learning clusters.

The results of the above queries are displayed in the contextual information field (number 4 in Figure 14) on the frontend. If a result is of particular interest, further detailed information can be viewed by clicking on the contextual cases that are shown in a popup window.

Contextual information about concessions is based on whether the issue is related to a concession or to a line item.

For concessions there are four queries to support the user:

- Part number query provides information about other concessions related to the same part.
- Description query provides information about other concessions that were similarly described (using NLP algorithms).
- Plant query finds concessions from the same plant as a certain concession. Plant-specific concessions can thus be managed and resolved more quickly.
- Supplier query provides concessions with the same part supplier as the current concession. Certain trends or characteristics may be identifiable of individual suppliers and their related concessions.

If a concession is related to line items, they can be selected for further information and for a k -nearest-neighbours query:

- K -nearest-neighbours query is based on the machine learning model as mentioned in Section 6. The query result shows the most similar line items, and is visualised in a graph (upper right corner of Figure 15). The graph helps in judging where the actual measurement lies and what decisions were made previously. Hence, saving time and efforts during the decision making process.

8 Demonstrator evaluation

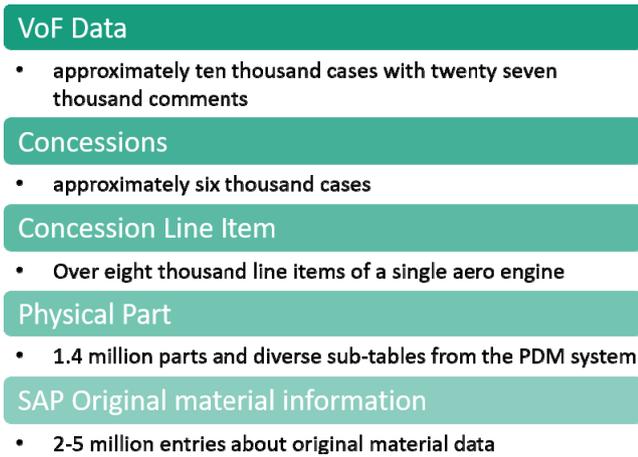
The development was conducted in agile sprints. This led to the evaluation of the demonstrator with the VoF team two times with an interval of approximately six months between them. The evaluation was done through a combination of a qualitative interview with feedback and followed by a usability study questionnaire. The interviews helped in capturing the positives and negatives of the demonstrator. The negatives and further improvement suggestions enabled the development of the demonstrator to suit the user requirements further. The questionnaire was used to capture the Perceived Usefulness, Perceived Ease of Use and User Acceptance (Davis, 1989).

The summary results of the questionnaire for the VoF case are: Perceived Usefulness with 4.6, Perceived Ease of Use with 3.9 and User Acceptance with 5 out of 5. The concessions use case implementation was evaluated twice with an interval of two months between each evaluation. Two members of the team were involved for both evaluations. The combined ratings on the questionnaire for the three categories were Perceived Usefulness with 4.56, Perceived Ease of Use with 4.2 and User Acceptance with 4.75 out of 5.

In addition, a performance test was conducted with the concessions team to measure the time taken to carry out a couple of tasks on the demonstrator and on their existing system. The tasks included finding an open concession, its relevant information, historic information of similar cases and making a decision about the open case. With the demonstrator, the decision about the case was made in approximately two minutes and thirty seconds, whereas with the existing system it took approximately four minutes for the same case. The users also expressed the benefit of saving time, and suggested integration of further queries to automate daily tasks.

The results show that users perceive the demonstrator to be useful, thus it has a high user acceptance. As it is still a demonstrator, the features are not fully implemented to the preferences of the users. Hence, there is room for improvement with respect to the ease of use. In addition, the feedback from the users indicated a possibility to use the same concept by different teams. An example given was for service engineers to manage service issues.

The demonstrator has also proven to be scalable as five major data sources were integrated into it, see Figure 16 for further numbers. The ontology has also been extended with the new data sources.

Figure 16 Data integrated into the demonstrator

Even with an increasing number of data sources throughout the development of the demonstrator, it has reacted in a robust manner. This is despite the significantly larger ontology queries and machine learning models.

9 Findings

For the research questions defined in Section 3, the project delivered initial evaluated answers. For RQ1, which looks for solutions to integrate heterogeneous data, the approach followed to clean and transfer the data into an intermediate database was the first solution. This was complemented by integration of an enterprise service bus which enables direct access of data in the desired format. The advantages of ontology mapping enable connecting enterprise data with the semantic middleware layer.

Section 6 describes different ways machine learning algorithms can be used to provide value-added services to the users which provides answers to RQ2. A larger data set and evaluations from the end-users assisted in selecting suitable algorithms and also to improve their accuracies. However, a need to further standardise options for free text fields was identified as an improvement for future processes. This would further help machine learning algorithms to improve their accuracies.

The complete system and research approach followed in the project illustrates how context relevant data can be generated and presented to the user, which is relevant for RQ3. The integration of the end-user in the process of development is crucial. Discussing with end-users possessing varying degree of experiences provides useful insights about the target group. The end-users assisted in defining the requirements for such a semantically connected system. The benefits of connecting different data sources and providing them with contextualised data were found to be advantageous in carrying out daily tasks. Furthermore, providing information through a centralised ontology provides flexibility for the development of new features on the existing software stack. This is relevant, as an industrial environment is dynamic and processes evolve with time. A modular and flexible software stack enables

customisation with reduced effort, leading to quicker implementation. The software stack was selected and designed to suit the needs of multiple applications and end users. It was also observed that the same architecture could be used for multiple use cases, hence making it reusable. Therefore, an information system based on ontologies can be used as a tactical tool to bridge the time until the enterprise systems are updated or even as a strategic information platform.

The system also encouraged end users to suggest creative front end features, which would help them carry out tasks quicker by using contextualised data in suitable graphical representations.

10 Conclusions

This paper presents the systematic design and development of a prototypical system to integrate semantic technologies into existing industrial infrastructure. In addition, the advantages of machine learning implementation on such applications are also discussed. The system supports functionalities such as interconnection of heterogeneous data sources, contextualisation of data for end users applications and adaptation to existing IT-infrastructure. There are currently efforts being made to further integrate the system into the work environment and current processes (at Rolls-Royce Deutschland). Owing the flexibility of the architecture, it is possible to integrate the system using various methods and tools. Some of the options are deploying it as a dedicated on-premise IT solution, running it through a cloud platform or by customising existing PLM platforms such as Teamcentre. One of the information system's key benefits is the ability to rapidly integrate information from a number of bespoke IT systems and link them to enterprise data. This enables an early realisation of business benefits while a strategic update of the data infrastructure is being developed.

The presented results are a first step towards bringing data and information from various data sources within design and production together. The method enables Rolls-Royce to connect information from design, production and aftermarket and – as a result – to improve design, product characteristics as well as production method. The influence of production variation on the final product characteristics can now be better described, and production methods can be optimised understanding their impact on the design and in-service behaviour. However, integrating such new developments into the daily activities of designers and manufacturing engineers requires further planning and careful integration of these capabilities into existing workflows. Both, processes and toolsets will need to be adapted to enable the improvements for all potential users.

The project opened up additional research questions such as consideration of a more diverse set of heterogeneous data sources, applying machine learning directly on ontologies and testing of scalability of the system with a larger ontology to name a few.

With further digital transformation of the business, a wider source of data and systems will become available requiring

more integration. Additionally, the implementation of interpretation methods, evaluation methods and potential chatbot systems will drive the future development of semantic information systems. The future challenge towards a broader implementation of true systems engineering consists of understanding the variety of available data (structured and unstructured) and correlating such data and information.

Acknowledgements

Cockpit 4.0 (CustOmised Collaborative Knowledge Pilot for Industrial Technology) is a collaboration project between Rolls-Royce Deutschland, Fraunhofer Institute for Production Systems and Design Technology and Brandenburgische Technische Universität Cottbus-Senftenberg. It is funded by the European Fund for Regional Development (EFRE) and by the state of Brandenburg.

References

- Adams, T., Dullea, J., Clark, P., Sripada, S. and Barrett, T. (2000) 'Semantic integration of heterogeneous information sources using a knowledge-based system', *Proceedings of the 5th International Conference on Computer Science and Informatics*, pp.1–6.
- Alegre, U., Augusto, J.C. and Clark, T. (2016) 'Engineering context-aware systems and applications: a survey', *Journal of Systems and Software*, No. 117, pp.55–83.
- Ast, M., Glas, M. and Roehm, T. (2014) *Creating an Ontology for Aircraft Design*, Deutsche Gesellschaft für Luft- und Raumfahrt-Lilienthal-Oberth, Report, pp.1–11.
- Bruno, G., Antonelli, D. and Villa, A. (2015) 'A reference ontology to support product lifecycle management', *Procedia CIRP*, Vol. 33, pp.41–46.
- Bruno, G., Korf, R., Lentjes, J. and Zimmermann, N. (2016) 'Efficient management of product lifecycle information through a semantic platform', *International Journal of Product Lifecycle*, Vol. 9, No. 1, pp.45–64.
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B. and Varoquaux, G. (2013) *API Design for Machine Learning Software: Experiences from the Scikit-Learn Project*. <http://arxiv.org/pdf/1309.0238v1> (accessed on 1 September 2020).
- Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M. and Xiao, G. (2016) 'Ontop: answering SPARQL queries over relational databases', *Semantic Web*, Vol. 8, No. 3, pp.471–487.
- Campello, R.J.G.B., Moulavi, D. and Sander, J. (2013) 'Density-based clustering based on hierarchical density estimates', *Proceedings of the Advances in Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, pp.160–172.
- Cruz, I.F. and Xiao, H. (2005) 'The role of ontologies in data integration', *International Journal of Engineering intelligent systems for electrical engineering and communications*, Vol. 13, No. 4, pp.245–252.
- Davis, F.D. (1989) 'Perceived usefulness, perceived ease of use, and user acceptance of information technology', *Management Information Systems Quarterly*, Vol. 13, No. 3, pp.319–340.
- Eclipse RDF4J (2020) *Eclipse RDF4J*. Available online at: <https://rdf4j.org/about/> (accessed on 1 September 2020).
- Ekaputra, F.J., Serral, E., Kiesling, E. and Biffel, S. (2017) 'Ontology-based data integration in multi-disciplinary engineering environments: a review', *Open Journal of Information Systems (OJIS)*, Vol. 4, No. 1, pp.1–26.
- Erkoyuncu, J.A., del Amo, I.F., Dalle Mura, M., Roy, R. and Dini, G. (2017) 'Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality', *CIRP Annals*, No. 66, pp.465–468.
- Gogineni, S., Exner, K., Stark, R., Nickel, J., Oeler, M. and Witte, H. (2019) 'Semantic assistance system for providing smart services and reasoning in aero-engine manufacturing', in Garoufallou, E. et al. (Eds): *Metadata and Semantic Research*, Springer International Publishing, Cham, pp.90–102.
- Grangel-González, I. (2019) *A Knowledge Graph based Integration Approach for Industry 4.0*, Dissertation (PhD thesis), Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany.
- Handler, M. (2010) *Semantic Data Integration in (Software+) Engineering Projects*, Diplomarbeit, Technische Universität Wien, Wien, Austria.
- Holler, M., Uebernickel, F., Brenner, W. and Uebernickel, F. (2016) 'Challenges in product lifecycle management – evidence from the automotive supply industry', *Proceedings of the 27th Australasian Conference on Information Systems*, Winterthur: ZHAW Zürcher Hochschule für Angewandte Wissenschaften.
- Kharlamov, E., Hovland, D., Jiménez-Ruiz, E., Lanti, D., Lie, H., Pinkel, C., Rezk, M., Skjæveland, M.G., Thorstensen, E., Xiao, G., Zheleznyakov, D. and Horrocks, I. (2015) 'Ontology based access to exploration data at Statoil', *International Semantic Web Conference*, pp.1–17. Doi: 10.1007/978-3-319-25010-6_6.
- Kusner, M.J., Sun, Y., Kolkin, N.I. and Weinberger, K.Q. (2015) 'From word embeddings to document distances', *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, pp.957–966.
- Liu, W., Zeng, Y., Maletz, M. and Brisson, D. (2010) 'Product lifecycle management: a survey', *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, ASME, San Diego, USA, pp.1–13.
- Louhdi, M.R.C., Behja, H. and El Alaoui, S.O. (2013) 'Transformation rules for building Owl ontologies from relational databases', *Proceedings of the 2nd International Conference on Advanced Information Technologies and Applications*, AIRCC Publishing Corporation, pp.271–283.
- Manaa, M.E. and Abdulameer, G. (2018) 'Web documents similarity using k-shingle tokens and Minhash technique', *Journal of Engineering and Applied Sciences*, Vol. 13, pp.1499–1505.
- McInnes, L., Healy, J. and Melville, J. (2018) 'UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction'. <http://arxiv.org/pdf/1802.03426v2> (accessed on 1 September 2020).
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) 'distributed representations of words and phrases and their compositionality', *Advances in Neural Information Processing Systems*, Vol. 26, pp.3111–3119.
- Mizoguchi, R. (2003) 'Tutorial on ontological engineering: part 1: introduction to ontological engineering', *New Generation Computing*, Vol. 21, No. 4, pp.365–384.

- Moghaddam, M., Cadavid, M.N., Kenley, C.R. and Deshmukh, A.V. (2018) 'Reference architectures for smart manufacturing: a critical review', *Journal of Manufacturing Systems*, No. 49, pp.215–225.
- Mountantonakis, M. and Tzitzikas, Y. (2019) 'Large scale semantic integration of linked data: a survey', *ACM Computing Surveys*, Vol. 52, No. 5, pp.1–40.
- Nemer, J. (2019) *Advantages and Disadvantages of Microservices Architecture*, Cloud Academy Inc. Available online at: <https://cloudacademy.com/blog/microservices-architecture-challenge-advantage-drawback/> (accessed on 31 August 2020).
- Niwattanakul, S., Singthongchai, J., Naenudorn, E. and Wanapu, S. (2013) 'Using of Jaccard coefficient for keywords similarity', *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, Hong Kong, pp.380–384.
- Otte, J.N., Kiritsi, D., Ali, M.M., Yang, R., Zhang, B., Rudnicki, R., Rai, R. and Smith, B. (2019) 'An ontological approach to representing the product life cycle', *Applied Ontology*, Vol. 14, No. 2, pp.179–197.
- Pan, J.Z., Vetere, G., Gomez-Perez, J.M. and Wu, H. (Eds.) (2017) *Exploiting Linked Data and Knowledge Graphs in Large Organizations*, Springer International Publishing, Switzerland.
- Peffer, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007) 'A design science research methodology for information systems research', *Journal of Management Information Systems*, No. 24, pp.45–78.
- Peruzzini, M., Mengoni, M. and Germani, M. (2012) 'PLM benefits for networked SMEs', *Proceedings of the 8th International Conference of Product Lifecycle Management*, Eindhoven, Netherlands, pp.205–218.
- Ren, Y., Jiang, L., Bu, F. and Cai, H. (2012) 'Rules and implementation for generating ontology from relational database', *Proceedings of the 2nd International Conference on Cloud and Green Computing*, IEEE, Piscataway, NJ, pp.237–244.
- Rousseeuw, P.J. (1987) 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, Vol. 20, pp.53–65.
- Roy, R., Stark, R., Tracht, K., Takata, S. and Mori, M. (2016) 'Continuous maintenance and the future – foundations and technological challenges', *CIRP Annals*, No. 65, pp.667–668.
- Staab, S. and Studer, R. (2009) *Handbook on Ontologies*, 2nd ed., Springer, Berlin, Heidelberg.
- Stuckenschmidt, H. (2011) *Ontologien. Konzepte, Technologien und Anwendungen*, 2nd ed., Springer, Berlin, Heidelberg, Berlin, Heidelberg.
- Studer, R., Benjamins, V.R. and Fensel, D. (1998) 'Knowledge engineering: Principles and methods', *Data & Knowledge Engineering*, Vol. 25, Nos. 1/2, pp.161–197.
- Usman, Z., Young, R.I.M., Chungoora, N., Palmer, C., Case, K. and Harding, J. (2011) 'A manufacturing core concepts ontology for product lifecycle interoperability', in van Sinderen, M. and Johnson, P. (Eds): *Enterprise Interoperability*, Springer, Berlin, Heidelberg, pp.5–18.
- Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S. and Gagnon, M. (2002) 'Ontology-based integration of information - a survey of existing approaches', *Proceedings of IJCAI-01*, pp.108–117.
- Wong, A. (2018) 'Introduction to model trees from scratch', *Towards Data Science*. Available online at: <https://towardsdatascience.com/introduction-to-model-trees-6e396259379a> (accessed on 1 September 2020).
- Zammit, J.P., Gao, J. and Evans, R. (2017) 'The challenges of adopting PLM tools involving diversified technologies in the automotive supply chain', in Rios, J. et al. (Eds): *Product Lifecycle Management and the Industry of the Future*, Springer International Publishing, Cham, pp.59–68.
- Zeid, A., Sundaram, S., Moghaddam, M., Kamarthi, S., Tucker, M. and Marion, T. (2019) 'Interoperability in smart manufacturing: research challenges', *Machines*, Vol. 7, No. 2, pp.1–17.