TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Dissertation

Accessing Manufacturing Data Through Virtual Knowledge Graphs

On the Value and Semantic Peculiarities of Time Series Data

carried out for the purpose of obtaining the degree of Doctor technicae (Dr. techn.), submitted at TU Wien, Faculty of Mechanical and Industrial Engineering, by

Benjamin Mörzinger

Mat.Nr.: 0926306

under the supervision of

Univ.Prof. Dipl.-Ing. Dr. Burkhard Kittl

Institute of Production Engineering and Photonic Technologies, E311

Reviewed by

Prof. Diego Calvanese
Free University of Bolzano
Dominikanerplatz, 3, 39100 Bozen,
Italy

Prof. Thomas Runkler
Siemens AG
Otto-Hahn-Ring 6, 81739 München,
Germany

Balanced Manufacturing (840746) and the Marietta Blau Grant
supported this work

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Vienna, 13.05.2019

Signature

Kurzfassung

Das Ziel, die Produktivität von industriellen Fertigungsanlagen durch den Einsatz von Informationstechnologie zu erhöhen und dadurch Hochlohnländer im globalen Wettbewerb zu stärken steht im Mittelpunkt von Initiativen wie „Industrie 4.0“ und „Smart Manufacturing“.

Im Zuge dieser selbsternannten Revolutionen versuchen fertigende Unternehmen unter anderem, den Erfolg von datengetriebenen Geschäftsmodellen auf Basis von Datenanalyse nachzuahmen. Die Idee, dass verborgene Informationen und Wissen in den Fertigungsdaten zu finden sind, führte zu einer schnell wachsenden Anzahl „intelligenter“ Messgeräte in modernen Fertigungsanlagen. Trotz eines anhaltenden Hypes und erheblicher Anstrengungen seitens Wissenschaft und Industrie sind konkrete Ergebnisse vor allem im nichtwissenschaftlichen Umfeld jedoch immer noch vergleichsweise rar.

Das Ziel dieser Arbeit ist es, diese Lücke zwischen angenommenem Potential von Fertigungsdaten und ihrer tatsächlichen Nutzung zu untersuchen und zu überwinden. Hierfür werden zwei Anwendungsszenarien aus industriellen Forschungsprojekten analysiert und bearbeitet. Es stellt sich heraus, dass nicht die Datenanalyse selbst, sondern vielmehr Datenzugriff der limitierende Faktor für die breitere Anwendung in der Fertigungstechnik ist. In Verbindung damit bringt die Wichtigkeit zeitlicher Daten in diesem Bereich weitere Komplikationen mit sich. Basierend auf dieser Analyse wird ontologie-basierter Datenzugriff (OBDA) vorgestellt, um einige der erkannten Herausforderungen zu überwinden. Um diesen Ansatz zu validieren, wurde eine Proof-of-Concept Implementierung entwickelt und mittels qualitativen Interviews mit Benutzern bewertet.

Weiters wurde ein umfassenderer Prototyp entwickelt, der dazu genutzt wurde um den aktuellen Stand der Technik auf seine Anwendbarkeit hin zu überprüfen. Zu diesem Zweck wurde eine domänenspezifische, temporale Ontologie entwickelt, die insbesondere für den Zugriff auf von Sensoren generierte Messdaten zugeschnitten ist. Es zeigt sich, dass dadurch zwar einige grundsätzliche Probleme gelöst, viele wichtige Anwendungsfälle jedoch nicht abgebildet werden können.

Daher wird, ausgehend von den domänenspezifischen Anforderungen der aktuelle Stand der Forschung auf dem Gebiet von OBDA mit Aggregationen über komplexe Zeitfenster erweitert. Hierbei handelt es sich um einen neueren, intuitiveren Modellierungsansatz, der eine formale Sprache umfasst, die mit Intervallaggregatfunktionen, Allen’s Intervallbeziehungen und verschiedenen metrischen Beziehungen für Zeitreihendaten ausgestattet ist. Auf Basis dieser Sprache werden bisher nicht realisierbare Anwendungsfälle aus der Praxis präsentiert und deren Abbildbarkeit experimentell durch Analyse von Realdaten validiert.

Abstract

The goal of increasing the productivity of industrial manufacturing plants through the use of information technology and thereby strengthening high-wage countries in global competition is the focus of initiatives such as ”Industry 4.0” and ”Smart Manufacturing”.

As part of these self-proclaimed revolutions, manufacturing companies are, among other things, trying to emulate the success of data-driven business models based on data analytics. The idea that hidden information and knowledge can be found in manufacturing data led to a rapidly growing number of "smart" sensors in modern manufacturing facilities. Despite persistent hype and considerable efforts on the part of science and industry, concrete results are still relatively rare, especially in the non-academic environment.

This work aims to investigate and overcome this gap between the assumed potential of manufacturing data and its actual use. For this purpose, two application scenarios from industrial research projects are analyzed. It turns out that it is not data analysis itself, but rather data access that is the limiting factor for a broader application in the manufacturing domain. In connection with this, the importance of temporal data in this domain brings further complications. Based on this analysis, ontology-based data access (OBDA) is introduced to overcome some of the identified challenges. To validate this approach, a proof-of-concept implementation was developed and evaluated through qualitative interviews with users.

Furthermore, a comprehensive prototype was developed, which was used to check the current state of the art for its applicability. For this purpose, a domain-specific, temporal ontology was developed, which is tailored in particular for access to measurement data generated by sensors. It turns out that this solves some fundamental problems, but many essential applications still cannot be illustrated.

Therefore, based on the domain-specific requirements, the current state of research in the field of OBDA is extended with aggregations over complex time windows. This approach represents a novel, more intuitive modelling approach that includes a formal language equipped with interval aggregate functions, Allen’s interval relationships, and various metric relationships for time series data. Based on this language, use cases that used to be unfeasible to realise with previous tools are presented. Also, their feasibility is experimentally validated through the analysis of real-world manufacturing data.

Acknowledgements

First and foremost, I want to thank my supervisor Prof. Dr. Burkhard Kittl for his calm, patient and knowledgable support of my studies. It was a pleasure and honour to be allowed to contribute to his decade-long effort of bridging the gap between manufacturing and computer science.

I also want to thank my colleagues at the Institute for Production Engineering and Photonic Technologies at TU Wien. Of all the names I would have to mention here, I want to specifically point out Thomas Weiler, who more then anybody else embodies the feedback, discussion and friendship I had the privilege of experiencing throughout my time at TU Wien.

Furthermore, I want to thank my colleagues from the KRDB group at Free University Bolzano for their hospitality. I could not have been more happy with the environment they provided me when I stayed with them. Their inputs and our cooperation significantly improved the quality of this thesis. I want to particularly point out Elem Güzel Kalaycı, who, despite being in the final phase of her own thesis, helped me wherever she could.

Similar to my colleagues from Bolzano, I want to thank my colleagues from the University of London. Without their inputs, I would not have had the language necessary to properly define my use cases. Especially Vladislav Ryzhikov excelled at this point and was always open for my questions and inputs.

Lastly, I would like to thank my colleagues at Infineon Austria AG for their motivation, knowledge and kindness which where a significant factor for the success we had in the BaMa project. I am grateful for their trust and fairness which enabled me to make many experiences I could not have made otherwise. I want to explicitly mention Karsten Buchholz in this respect, who always was open for my ideas. Especially because most of them directly translated into more work for him.

As my friends and family are concerned, it is not possible to comprehend the support I received from them. Nevertheless, I want to mention the person that certainly influenced this work the most, which has to be Christina Gahn. Our conversations on research and science greatly helped me with my understanding of the topics and consequently significantly improved the structure of this work.

Thank you all.

Publications

Listed here are the papers that we wrote and that have been published or have been accepted for publication during this PhD study.

Journal Articles

- Christoph Zauner, Florian Hengstberger, Benjamin Mörzinger, Rene Hofmann, and Heimo Walter, "Experimental characterization and simulation of a hybrid sensible-latent heat storage," *Applied Energy*, vol. 189, pp. 506–519, 2017.
- Ines Leobner, Peter Smolek, Bernhard Heinzl, Philipp Raich, Alexander Schirrer, Martin Kozek, Matthias Rössler, Benjamin Mörzinger, "Simulation-based Strategies for Smart Demand Response," *Journal of Sustainable Development of Energy, Water and Environment Systems*, vol. 6, no. 1, pp. 33–46, 2017.
- Peter Smolek, Ines Leobner, Georgios Gourlis, Benjamin Mörzinger, Bernhard Heinzl, and Karl Ponweiser, "Hybrid Building Performance Simulation Models for Industrial Energy Efficiency Applications," *Journal of Sustainable Development of Energy, Water and Environment Systems*, vol. 6, no. 2, pp. 381–393, 2018.

Conference Publications

- Benjamin Mörzinger, Thomas Weiler, Thomas Trautner, Iman Ayatollahi, Bernhard Angerer, and Burkhard Kittl, "A large-scale framework for storage, access and analysis of time series data in the manufacturing domain," in *Procedia CIRP*, vol. 67, pp. 595–600, 2017
- Benjamin Mörzinger, Christoph Loschan, Florian Kloibhofer, and Friedrich Bleicher, "A modular, holistic optimization approach for industrial appliances," in *Procedia CIRP*, vol. 79, pp. 551–556, 2019.
- Alexander Raschendorfer, Benjamin Mörzinger, Eric Steinberger, Patrick Pelzmann, Ralf Oswald, Manuel Stadler, Friedrich Bleicher, "On IOTA as a potential enabler for an M2M economy in manufacturing," in *Procedia CIRP*, vol. 79, pp. 379–384, 2019.

- Peter Smolek, Georgios Gourlis, Benjamin Mörzinger, Ines Leobner, and Karl Ponweiser, "Thermal Building Simulation of an Industrial Production Facility using the Balanced Manufacturing Approach," in *Proceedings of 12th Conference on Sustainable Development of Energy, Water and Environment Systems*, 2017.

Posters and Workshops

- Benjamin Mörzinger, Martha Sabou, Fajar J. Ekaputra, and Nikolaus Sindelar "Improving industrial optimization with Semantic Web technologies," in *CEUR Workshop Proceedings*, vol. 2198, p. 4., 2018

'. . . There's glory for you!'
'I don't know what you mean by GLORY*,' Alice said.*
Humpty Dumpty smiled contemptuously. 'Of course you don't – till I tell you. I meant "there's a nice knock-down argument for you!"'
'But GLORY *doesn't mean "a nice knock-down argument,"' Alice objected.*
'When I use a word,' Humpty Dumpty said, in rather a scornful tone, 'it means just what I choose it to mean – neither more nor less.'
'The question is,' said Alice, 'whether you can make words mean so many different things.'
'The question is,' said Humpty Dumpty, 'which is to be master – that's all.'

exchange between Humpty Dumpty and Alice,
Through the Looking-Glass by Lewis Carroll

Contents

Chapter 1

Introduction

Science, as Wilson [178] defines it, is the organised, systematic enterprise that gathers knowledge about the world and condenses the knowledge into testable laws and principles. Scientists formulate research questions based on their observations. They develop hypotheses which are tested, using suitable methods. Apart from formal sciences such as mathematics, results can either falsify or support, but never prove hypotheses.

In the case of manufacturing, we seek to improve the ways we create physical objects. This can be achieved in many ways. New processes could make it possible to manufacture products in less time, improved processes need fewer resources (e.g. human labour, time, energy) or entirely novel products can be created. Researchers in the manufacturing domain often chose experimental investigation as a method. Through experiments, defined processes are surveilled under specific conditions using sensors. These sensors create data, which is analysed to build models of the real world. Such models might, for example, describe the relationship between different physical quantities, product quality, tool wear or emissions.

Consider, for example, the following scenario: A milling machine is used to manufacture a product. After some hours, the product quality decreases significantly. There are no apparent reasons for this, and the phenomenon only appears on one particular machine. Consequently, a potential research question is: "What are the reasons for the quality deterioration?". Responsible engineers might suspect, that the production processes lead to a temperature increase within the machine, which reduces its manufacturing precision (Hypothesis). A feasible way to investigate this question might be the following experiment. Temperature sensors are deployed to the machine in question and another, equivalent one. Then, a series of defined production processes are scheduled, and the resulting temperature signals stored. Through a comparison of those signals, the hypothesis is tested. If, after some time, the temperature within the machine under question is significantly higher then within the other one, the hypothesis is supported. However, other causes can never be ruled out, and further investigation might be necessary. For example, in a second step, quality criteria for the generated products need to be measured. Only if a significant deterioration, which also correlates with the temperature increase, exists, the hypothesis is not falsified.

Experiments like this might seem a bit hypothetical, but in today's modern, industrial production facilities every single machine tool has hundreds of sensors such as the temperature sensors from the example deployed to it. These might be position, velocity and acceleration sensors connected with the machines Numerical Control (NC), but also microphones or force sensors. Due to initiatives such as "Industrie 4.0" and "Smart Manufacturing", which promote digitisation of industrial manufacturing, the number of such data generating devices steadily increased and with this the data that they generate. As a consequence, the manufacturing domain today advanced to be one of the most data-intensive domains with approximately two exabytes of newly stored data in 2010 [130]. The supposed potential of that data is heavily promoted. So indeed, this claim seems reasonable. Imagine the experiment above to take place not in isolation, but within a modern production facility. Only a small subset of deployed sensors (the temperature sensors) is used to test a hypothesis. The remaining sensors, however, continue to create data throughout the experiment. The number of *potential* research questions, which is the total number of possible unordered combinations of size k from the total number of available sensors n, can be calculated using the following equation:

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!} \tag{1.1}$$

Assumed, that approximately five sensors are required to answer a typical research question ($k = 5$), and each machine has one hundred sensors deployed to it ($n = 100$), there exist 75.287.520 potential *other* research questions, which could be answered through analysis of data that is generated throughout one single experiment. The question is: given this number, why is it still necessary to do experiments in manufacturing engineering? As will be shown, this is not due to inadequate data analysis tools, but rather a consequence of a much more delicate problem: limited data access.

1.1 Motivation

The idea of using data to answer questions is anything but new. Relatively recently, due to advances in Information Technology (IT), however, terms such as machine learning, data mining, big data or artificial intelligence along with the success of data-driven enterprises have brought the attention of a larger audience back to the elusive beast which is referred to as data. As described, data on its own is not yet useful. Also, it is not "the new gold". As any message board on the internet shows, data can be generated without much effort and only small parts of it, if anything, contain knowledge or even useful pieces of information. In order to find these, data needs to be processed. For data that is stored in databases, this is called Knowledge Discovery in Databases (KDD). According to Fayyad and Uthurusamy [63], KDD is "the non-trivial process of identifying valid, potentially useful and ultimately understandable patterns in data". The notion to computationally analyse data to acquire new knowledge fundamentally changed the way professions such as marketing, retail, and politics work. Also in science, medicine, biology, and physics, KDD is used intensively.

Also in manufacturing, a large number of potential application scenarios supposedly exist. KDD is used for things such as factory energy monitoring [146], prediction of manufacturing power consumption [160], lead time estimation [138] or dispatching rule discovery [119] to name just a few. This makes the fact that there are still relatively few large-scale implementations of KDD techniques to be found in the industry even more intriguing. To investigate this further, it is necessary to have a closer look at the KDD-process, which in literature is defined as follows:

1. Data Selection: Selecting the appropriate dataset and accessing the correct subsets. Often, this also involves joining data from different, heterogeneous data sources, which requires significant implementation efforts.

2. Preprocessing: Both cleaning and filtering of data, are carried out to improve the quality of the data set and therefore the potential results. This is a non-trivial task, as it might not be obvious what parts of the data can be considered outliers, noise or erroneous.

3. Transformation: This step involves reduction and projection of cleaned data into meaningful features to reduce the number of considered variables to a minimum.

4. Data Mining: Depending on the goals and the data, appropriate methods and algorithms are applied to process the transformed dataset. This generates a model describing the data. This often is an iterative process, and it might be necessary to apply several different methods sequentially to reach the desired goals.

5. Interpretation: A significant part of this step is the visualisation of results. Often, the interpretation leads to the understanding that to reach the stated goals, the whole process needs to be repeated.

This process definition is very specific and does not cover a crucial step: problem formulation. This step is required before data selection and would, in a scientific context, result in the formulation of a research question. Therefore, in this thesis we will use a more general model of the KDD process, which is shown in Figure 1.1. Here, data selection and preprocessing are joined to form the access phase. Transformation, data mining and interpretation are combined to form the analysis phase. For a more detailed description, the reader is referred to Chapter 4.

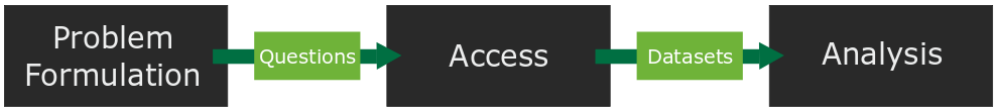


Figure 1.1 Data science process according to Kharlamov et al. [100]

Out of the steps, scientists seem to find *analysis* as the most interesting one. Much research focuses on the development of new algorithms or gradual improvement of existing ones as well as data visualisation and other techniques to improve the interpretation of large data sets. The goal

here often is to improve further the quality of models derived for a wide variety of application areas and with it, a wide range of data. In some domains, however, the analysis might not be the most limiting factor. Based on their investigations of large scale, industrial KDD scenarios, Kharlamov et al. [100] argue that up to 80% of the total time needed to finish the data science process is spent on accessing the required data. This is, according to them, due to inconsistent documentation, changing naming conventions and communication problems. Among other factors, this prevents data-driven approaches from being applied in the industry. Harding et al. [83] support this argument. In their work, they describe application scenarios, but also address future research fields. Among others, they name (data) preparation issues and explicitly highlight significant effort that is needed to prepare data for processing as one of the limiting factors for the broader application of data mining in the manufacturing domain. Also, they mention the need to "enhance the expressiveness of knowledge" for rules generated by data mining.

On the technological side, data access today is solved through Relational Database Management System (RDBMS). Those systems are designed to abstract the physical storage of data away from users and make it accessible via an additional, logical layer. This abstraction has led to significant developments and made the way (human) agents interact with data much more effective. Throughout the past decades, the knowledge in the field of mathematical logic has developed dramatically and led to what is sometimes referred to as the "logical revolution" [172]. In the course of this "revolution", several reasoning systems were developed in order to (also) simplify the way databases can be queried. This investigation makes the apparent lack of accessibility to data especially remarkable.

Abiteboul et al. [2] name the enrichment of data management with knowledge as one of the leading research directions in data management. According to the authors, purely relational database management systems today face limitations because that data is stored in distributed sources and in general cannot be considered to be complete. This significantly increases the challenge that access to data poses today. To ease this, the authors propose to use available knowledge to aid interaction with stored data. To do this, they propose techniques from Knowledge Representation, which is a sub-area of Artificial Intelligence.

As Benedikt and Michael [22] point out, reasoning systems are a promising way to simplify database querying, but more emphasis needs to be put on the practical application of techniques that have been developed in the past decades. Consequently, in this thesis, the application of a particularly promising approach to knowledge enriched data access, namely Ontology-based Data Access (OBDA) will be investigated from the perspective of manufacturing engineering as a potential application area. As illustrated in Figure 1.2, OBDA uses a knowledge base in the form of an ontology to aid and improve queries over data that is stored in conventional RDBMS [41]. Ontologies reflect the view of experts on their domain, which leads to a much more natural way of query formulation which does not require any knowledge about the schema of the data source. Furthermore, it allows to enrich (saturate) queries based on knowledge. To make this possible, the ontology is connected with the data source through mappings. These mappings allow an OBDA system to translate saturated queries into

Structured Query Language (SQL)-queries, which can be processed by the database management system.

To illustrate this, consider the following example: A database is used to store data from temperature measurements within a baking oven. An ontology is used to describe, that some of these sensors correspond to a concept **OvenTemperature**, others to the concept **MeatTemperature**. Using an ontology, the facts that **OvenTemperature** and **MeatTemperature** are each special kinds of **Temperature** and that this quantity is normally measured in Kelvin can be stored in an explicit, machine-readable way. Independent from how the data is stored, such an ontology can now be used to reason that somebody who asks for **OvenTemperature** asks for a **Temperature**. Therefore, whatever the result of that query would be, it should be displayed with a suffix "K". Also, if a user were to ask for "all temperatures", the system would return measurements stored as either **OvenTemperature**, **MeatTemperature** or **Temperature** without the necessity for any further information from the user. Most importantly, this approach does not require the data to be altered in any way. Also, data does not need to be moved to an application specific intermediate storage location which would introduce redundancies which often end up to be inconsistencies. In the next section, we will further investigate open questions on how this idea could be translated to the manufacturing domain.

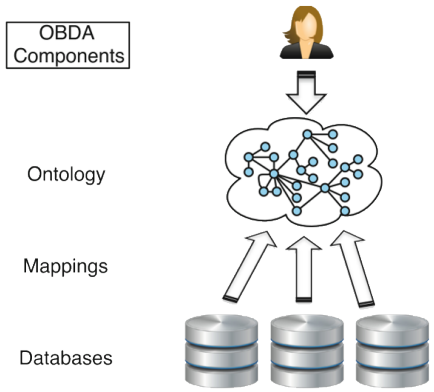


Figure 1.2 Conceptual illustration of OBDA [100]

1.2 Research Questions

There seems to be some degree of understanding among experts, that data access is the bottleneck within the data science process. Consequently, in 2012 a consortium of European research institutions and industrial companies started the OPTIQUE[1], which was funded by the European Union and aimed at facilitating industrial data science. Kharlamov et al. [100] report on one of their use cases which was provided by Siemens Energy. Service centres monitor appliances such as turbines, generators, and compressors operated and distributed globally. Engineers at those centres analyse data generated by sensors to investigate current malfunctions and provide countermeasures to service personnel on premise. The authors claim that data access accounts for 80% of the total time spent by engineers. They base this number on reports from practitioners at Siemens. A similar claim is made by Crompton [56] for the Oil and Gas industry. An in-depth investigation on this matter, however, seems to be missing.

[1] http://optique-project.eu/

Scenarios similar to the ones described by Kharlamov et al. [100] appear in other domains, such as manufacturing engineering as well. Deployed sensors generate data representing different aspects of the underlying processes. Depending on the exact application scenario, data is related to different physical quantities such as temperatures, forces or accelerations. Just as engineers at Siemens use monitoring data from turbines to predict future malfunctions, researchers in the manufacturing domain might interpret data generated by acceleration sensors or microphones to detect unstable processes. In the manufacturing domain, even though Harding et al. [83] mention an apparent data access problem, they do not elaborate on that. Recent works which try to enable data mining in the manufacturing domain [184, 169] do not mention data access at all. A shared understanding regarding the severity of this problem is, it seems, still missing.

More recently, similar observations regarding the problematic role of data access in data science processes were made in the research project Balanced Manufacturing (BaMa) [25][2]. Within this four-year research project, 18 partners developed software solutions aimed at helping industrial companies to operate their machines more efficiently. A multidisciplinary team of researchers defined the methodical basis for software tools developed by development partners. These tools were then implemented at industry partners from different production sectors (metalworking industry, Electronics industry, food industry) and evaluated. At their core, these tools build upon simulation models of the production plants in question, which in turn where used to anticipate the behaviour and find optimal solutions to changing scenarios. Data from monitoring devices had to be used to build those simulation models and improve their predictions [129]. The definition, access and analysis of these often large amounts of real-world industrial data proved to be an unforeseen challenge within the project. These experiences, which will be further discussed later, support the claim that data access is an issue for implementation of data science solutions in a wider range of industries. A detailed, quantitative analysis of the scope of and the reasons are, however, still missing. Detailed analysis of this question has not yet been carried out. Therefore, the following research question arises:

Research Question 1 (RQ1): *Can the claims of other authors regarding the severity of problems induced through data access be confirmed?*

Independent from the exact answer to the first research question, we are interested in how the existing limitations to data access could be overcome. According to Kusiak [113], today, there seems to be a general lack of suitable software solutions that properly enable the use of data in the manufacturing domain. Companies still operate in isolation and research does not address practical problems enough. Among the innovation gaps he identified is the need for improved data collection, use, and sharing. Kharlamov et al. [100] claim that OBDA is a suitable solution to overcome (at least parts) of this. Their documented record of industrial projects in this field supports this claim. If, and more importantly how these results can, however, be translated to the manufacturing domain, is still unclear.

[2] bama.ift.tuwien.ac.at

A few specific factors need to be considered, which distinguish the manufacturing domain from others. First, there is the problem of data sources. The majority of meaningful use cases, requires a combination of data from different sources such as Enterprise Resource Planning (ERP) or Manufacturing Execution Systems (MES). Consequently, any solution aimed at improving access to manufacturing data needs to be compatible with such systems. Currently, to enable data science, often intermediate data storages are created by domain experts through "grassroots solutions", which tend to be cumbersome. Excel sheets, Comma-Separated Values (CSV)-exports, and sometimes RDBMS are typically used "technologies" in this respect.

Another important class of data sources are production machines and sensors connected with them (**machine data**). This class of sources often is necessary for implementing exact data science application scenarios, but also introduce two significant problems. The first problem is limited access to that kind of data. Despite significant effort towards standardisation, today this data still is not easily accessible. In reality, the majority of machines generate data in vendor specific, often proprietary formats which leads to an often insurmountable number of individual solutions tailored to specific application scenarios, which typically can not interact with each other and only rarely can be transferred between different application scenarios.

The second problem is the temporal nature of machine data itself. Machine data typically is time series data, which introduces some complications which make both their facilitation for data science and their access through OBDA particularly hard [113]. Domain-specific requirements are a prerequisite to determine necessary features for any potential solution. Therefore, the second and third research questions are the following:

Research Question 2 (RQ2): *What are the manufacturing-domain specific system requirements that limit access to data?*

Research Question 3 (RQ3): *How can limitations to data access in the manufacturing domain be overcome?*

Based on the formulated requirements, a prototypical implementation of a potential solution is possible. The performance of any such prototype needs to be evaluated in order to determine whether or not it is feasible and to identify further necessary research steps. Consequently, the final research question that this thesis will try to answer is:

Research Question 4 (RQ4): *If applied, how efficiently can the identified solutions be applied to the identified problems in the manufacturing domain?*

1.3 Hypotheses

As already mentioned in the previous section, observations and informal talks with different stakeholders within the BaMa project indicate, that the amount of time that is spent on data access is significant.

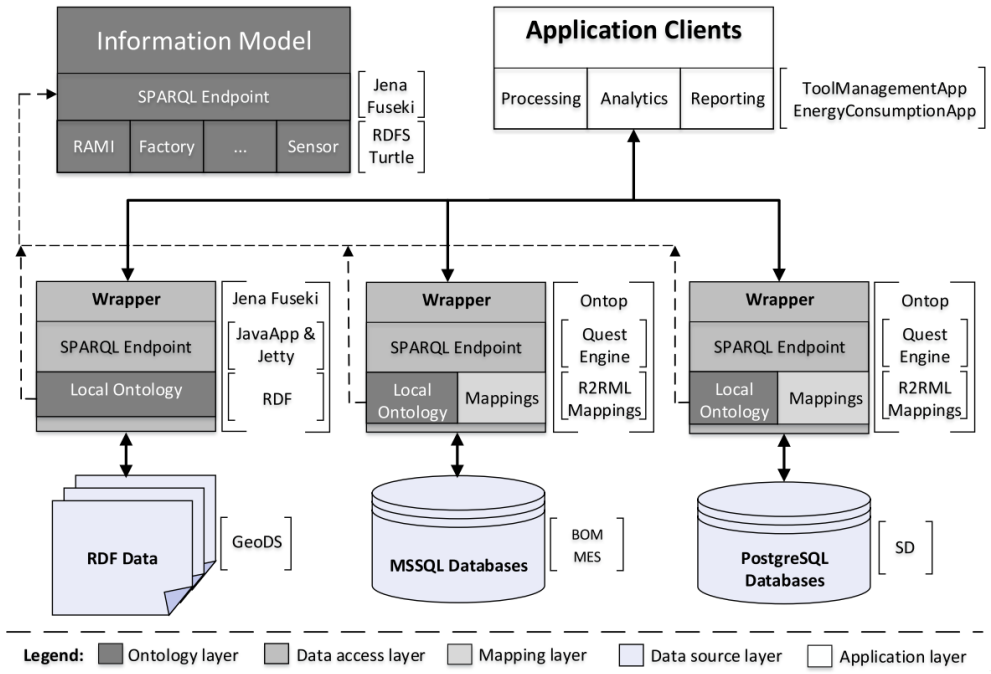


Figure 1.3 Integrated information model architecture for the manufacturing domain [142]

This would support the claims made by Kharlamov et al. [100]. Therefore, the hypothesis regarding research question 1 is the following:

Hypothesis 1 (H1): *A significant amount (>60%) of hours spent throughout the implementation process is spent on data access.*

It seems reasonable to say that data access is a problem in the manufacturing domain. This is due to heterogeneous data sources, and frequently changing requirements. This makes it hard to create and operate conventional software solutions which typically require fixed processes and consistent, interconnected data sources. In the case of the manufacturing domain, constantly and fast-changing customer demands require manufacturing systems, including software tools, to be flexible as well [127]. Furthermore, the lack of qualified personnel to take care of the data that is constantly generated and processed makes it hard to access generated data. As a result, Hypothesis 2 is formulated:

Hypothesis 2 (H2): *Data integration, access to time series data and lack of IT specific knowledge are the main limitations to efficient data access in the manufacturing domain.*

OBDA seems to be a promising technology for several domains, as the application to use cases have shown its practicality [100]. Regarding the application of this technology and Semantic Web Technologies in general, Petersen et al. [142] provide an architecture for an integrated information model for the manufacturing domain. They illustrate how data from distributed, heterogeneous

sources such as Bill of Material (BOM), MES or sensors can be integrated. The architecture that they provide, which is depicted in Figure 1.3 uses state of the art technology and is compatible with existing software systems as they facilitate OBDA, which makes it possible to use Semantic Web Technologies for data access while being able to keep the data stored in RDBMS.

Based on the nature of use cases that are typically dealt with in manufacturing related application scenarios of KDD, however, it can be expected that temporal aspects (i.e., aspects that are related to time) will also be an essential requirement from that area. The specific requirements imposed by this fact is only superficially touched in the work of Petersen et al. [142], which probably has to do with the fact that handling of temporal data still is an open research topic [179]. Regardless, Hypothesis 3 is:

Hypothesis 3 (H3): *OBDA is a suitable technology for improving data access in the manufacturing domain. In order to be a feasible choice as a solution for data access in the manufacturing domain, however, OBDA needs to provide solutions for accessing temporal data which are currently not available.*

If Hypothesis 3 cannot be falsified, the domain-specific requirements go beyond the state-of-the-art. As can be seen in Petersen et al. [142], current approaches to accessing time series data (which is generated by sensors and accounts for the major part of the data that is generated in the manufacturing domain) are rather basic and do not provide much expressivity. Consequently, existing technologies need to be extended. If it is possible to incorporate temporal concepts into OBDA, intuitive interaction of domain experts with time series data would be possible, which can be expected to lead to a significantly reduced time demand for data analysis tasks. Brandt et al. [29] propose to expand an existing OBDA framework with the capability to deal with temporal concepts to improve data access in similar settings. Consequently, Hypothesis 4 is the following:

Hypothesis 4 (H4): *Queries become considerably simpler if OBDA is applied. Furthermore, the overall time needed to answer queries does not increase significantly in comparison to conventional systems.*

1.4 Method

To test Hypothesis 1, the work that has been done on one of the application scenarios within the BaMa research project was analysed. As BaMa was a funded research project, detailed records regarding the working hours of each project partner were available. Some parts of those records included information regarding the specific activities that were carried out. Those activities were described in natural language which made it necessary to classify them. This was done manually by assigning tasks and task descriptions to the respective stages of the data science process introduced above. Based on this classification, the amount of time spent per class was calculated and analysed to derive the respective timeshare per process phase.

In order to identify domain-specific requirements (Hypothesis 2), a literature review regarding application scenarios in the manufacturing domain was conducted. Furthermore, throughout this thesis, two application scenarios are introduced and analysed concerning their requirements towards the data science process.

Based on the results of these steps, a set of requirements was defined and used as guideline for evaluating current OBDA research results in respect to the potentially necessary development steps in order to live up to those requirements. Therefore, in order to test Hypothesis 3, another literature review on the current state of OBDA was conducted. Then, a preliminary OBDA Proof of Concept (PoC) was developed based on the aforementioned application scenario from the BaMa project. Based on the lessons learned from this PoC, a prototype was developed based on the specific requirements from the second application scenario. As the basis for this more extensive OBDA- prototype, a database for manufacturing process related measurement data was implemented. This database was populated with data generated by more than 50 sensors deployed to a machining centre. This centre was used to investigate drilling processes experimentally. Different researchers can query data and carry out a variety of analysis tasks. In this light, currently available OBDA solutions were evaluated and compared with user requirements.

Lastly, in order to test Hypothesis 4, the developed prototype was evaluated based on the formulated requirements. Especially, representative, example data analysis tasks were used to compare necessary efforts with and without the use of this prototype. User queries and their conventionally generated SQL translations were compared to similar queries when an OBDA system is in place.

1.5 Contribution

The main contribution of this work is the bridging of another gap between two disciplines, namely computer science and mechanical engineering. There are two main reasons for this gap to prevail. First, manufacturing engineers typically do not have the necessary knowledge to identify and develop suitable IT solutions for their application scenarios or even formulate their requirements in a way that is understandable for computer scientists. Computer scientists, on the other hand, only rarely have the chance to be confronted with concrete application scenarios that fully use the tools that they have at hand.

The first contribution of this research is a detailed, quantitative analysis of the magnitude of current obstacles towards data science with a focus on the manufacturing domain. This analysis will help to motivate the development of software tools and might indicate some research directions for the future. For the specific case of data access, we identified and analysed requirements from the manufacturing domain perspective and compared this with existing tools.

Based on these requirements, a prototypical implementation of an OBDA tool for the manufacturing domain was developed. This provides two contributions. First, in the field of manufacturing, it will provide a framework for the integration of OBDA into existing manufacturing systems. In the

field of computer science, this work provides requirement specifications from concrete applications scenarios that motivate further development of temporal description logics.

This work showed the potential and limitations of this technology in a particular field where it has not been applied before. Motivated by requirements from additional application scenarios, the current state of research on the topic of OBDA was expanded by an ontology on manufacturing processes and sensory data generated by them. Furthermore, a more intuitive and general way to formally describe and reason over temporal data was introduced. Evaluation results, as the final contribution, give insights into the potential of the proposed solution and might motivate fruitful areas for optimisation.

1.6 Structure of this document

This thesis is structured as follows. In the following Chapter 2, the technological foundations for the work conducted in this thesis are described. These are the essential characteristics of time series data and how they interfere with the mining of that kind of data. Furthermore, foundations for databases and OBDA are given.

In Chapter 3, the state of the art will be reviewed. This entails current data mining endeavours in the manufacturing research community as well as an overview of current attempts to improve data access and integration in the manufacturing domain. Based on this, the defining aspects of OBDA will be described and current research efforts towards the inclusion of temporal aspects into OBDA will be reviewed.

In Chapter 4, an application scenario from the research project BaMa is described. The section provides a quantitative characterisation of the role of data access and the (supposed) severity of the problem in the particular case. It concludes with a PoC based on virtual knowledge graphs and its evaluation through user interviews.

In Chapter 5, another example use case is introduced. It is representative of the manufacturing domain in a sense that several different sensors are used to generate time series data that can be used to investigate manufacturing processes. Requirements of a corresponding data access system are laid out in the form of example queries and analysis tasks. In Chapter 6, the prototypical implementation of a prototype that accommodates the formulated queries, as well as the relevant development steps, is described. Furthermore, the prototype is evaluated through experiments, using data from conducted drilling processes. Finally, in Chapter 7, the stated research questions are answered based on the results of the conducted work. Also, I comment on the limitations of the developed solution. I close this thesis with an outlook on further research topics.

Chapter 2

Preliminaries

In the following sections, the fundamental notions that this thesis is building upon are laid out. These are description logics, database systems and OBDA.

2.1 Time Series Data

Depending on the background of the respective author, the term time series data can have somewhat different meanings. This ambiguity gives rise to considerable problems as what one domain considers as time series data might be much simpler then what another domain might be actually referring to. This is also true for the topics discussed in this thesis. Therefore, some explicit definitions are necessary to distinguish different kinds of time series data.

Time series data: Sometimes also referred to as temporal data. Any data that is stored digitally as a series of discrete timestamp- value pairs is considered time series data. Here, we do not distinguish between different data types that the value might have. A series of strings denoting NC-program names executed on a machine is just as much a time series as measurements of a sensor represented as a float value are. Different kinds of time series data can furthermore be distinguished according to the following two dimensions.

Historical vs Streaming Time series data: As streaming data, we consider any time series data that is updated while it is processed. This potentially introduces significant complications. In contrast, if updates do not need to be carried out in parallel to analysis tasks, we refer to historical or non-streaming data. Examples for streaming data might be alarms that trigger if a defined threshold is surpassed or Key Performance Indicator (KPI) that are visualised to give workers live feedback in a factory. In contrast, temperature measurements of the atmosphere or election results might be considered historical time series data.

Event logs vs time series logs: This second dimension is distinguished based on the nature of the underlying phenomenon. A series of discrete phenomena in time stored as time series is referred to as event log. In contrast, if the underlying phenomenon is continuous in nature, we refer to the resulting time series as time series logs. Examples for event logs are results of quality measurements

or executed active tools per machine. Time series of this type are generated through some event which triggers an update. The discrete nature of the underlying phenomena is a source of considerable simplification. In contrast, time series logs are time series which are created through time series of continuous processes. Examples for this kind of data might be data generated by a temperature sensor representing a room temperature or by a force sensor applied to a milling tool holder.

This thesis is concerned with historical time series data in the form of time series logs. Some of their specific peculiarities will be discussed in the following sections.

2.1.1 Time Series Logs

Time series logs are digital representations of continuous phenomena. As that, they have some features that are unique to them. Those features also influence how systems need to be built in order to be able to access this type of data. Continuous signals from measurements of physical quantities are especially crucial in natural sciences and engineering. In Fourier Analysis, any periodic function in time can be decomposed into a weighted sum of a (potentially infinite) set of simple trigonometric functions. Consequently, any periodic signal can, therefore, be understood as the superposition of sines and cosines with different amplitudes at different frequencies. For each periodic signal, in the time-domain, an equivalent signal in the frequency domain can be found. The transition between time and frequency domain is done through Fourier transformation. The distribution of amplitudes as a function of frequencies is referred to as the frequency spectrum of the signal.

Through sampling, such continuous signals can be converted to time series data. The conversion of continuous signals to discrete series, however, results in a loss of information. Intuitively, continuous signals have distinct values at any point in time. Discrete series only have values at time points where measurements where made. For any other time point, some assumption has to be made. A critical metric for describing sampling processes is the sampling rate, which is the distance between consecutive measurements. The higher this rate, the less information is lost in the sampling process.

For any given signal with limited bandwidth (the highest frequency f_{max} in the frequency spectrum of the signal is finite), a minimum sampling rate f_s is defined which is required to prevent the loss of information due to sampling. This criterium is known as the *Nyquist–Shannon sampling theorem* and defines the minimum sampling rate as $f_s = 2 \cdot f_{max}$. This means, that in order to be able to accurately reconstruct a continuous signal of frequency f_{max} from a sampled time series, the sampling rate must be at least double the highest frequency of the original signal.

As is the case for any measurement, also for the sampling of time series data errors will appear. Among those errors are value/quantisation errors, systemic errors and stochastic errors. Concerning the last error class, the notion of the frequency spectrum can be used to elaborate this further. What is to be considered noise depends on both the signal under consideration and the sensors that were used to create the signal and sample it. When dealing with real-world time series data, often this data needs to be cleaned from noise. For this, filters are used.

For the scope of this work, the general family of discrete linear filters will be relevant. Following the notation in [153], the output y_k of such a filter of order q is given in Equation 2.1a. The coefficients a_i and b_i are filter parameters which can be used to determine the properties of the filter. Based on the particular choice of parameters, different filter classes can be realised. For example, by setting all coefficients $a_i = 0$, only past elements of the time series are considered in the calculation of the output. A finite input, therefore always leads to a finite output of the filter, which is why this class of filters is called Finite Impulse Response (FIR)- filter. Another filter class can be created by setting $a_i = [0]$ and $b_i = [\frac{1}{q}, \ldots, \frac{1}{q}]$, which simplifies Equation 2.1a to Equation 2.1b. As can be seen, this results in a moving average over the past q values of the time series. This moving average filter is also known as FIR-low-pass and will be used in Chapter 6.

$$y_k = \sum_{i=0}^{q-1} b_i \cdot x_{k-i} - \sum_{i=1}^{q-1} a_i \cdot y_{k-i} \tag{2.1a}$$

$$y_k = \sum_{i=0}^{q-1} \frac{1}{q} \cdot x_{k-1} = \frac{1}{q} \sum_{i=k-q+1}^{k} x_i \tag{2.1b}$$

Examples of time series data in the manufacturing domain might be position measurements from a machine axis or the hydraulic oil temperature of a brake system. The analysis of such data can lead to a better understanding of the underlying processes or indicate irregularities.

Time series data has some characteristics that make handling them hard. Those are large data size, high dimensionality (number of data points) and, in the case of streaming data, continuous updates. Searching non-time series data in databases traditionally is carried out by finding exact matches to some criteria. For time series, however, this often is not feasible, as the concept of similarity is more delicate, as two time series that would be considered "similar" will hardly ever exactly be the same [66]. To answer queries of the form "find all power measurements which behave similarly to power measurement B", human perception of shape is highly effective. It relies on the shape of time series data (or rather their visualisations). Currently, however, available software tools are not able to mimic this ability sufficiently, which motivates research on the topic of similarity measures [62].

2.1.2 Issues with Time Series Data

From the perspective of time series data mining, two time series specific issues are especially relevant:

- Data representation: As already mentioned, the shape of time series data is a natural way to describe it. A suitable representation of time series also is the prerequisite for an appropriate reduction of dimensionality.
- Similarity measures: The task to compare (at least) two pieces of data with each other is fundamental for data mining. This requires a way to determine if two things are similar to each other. Similarity measures are used to achieve this.

Time series data representation has the goal to reduce the dimensionality of time series data while still retaining its characteristics, so that time series that are found to be similar before the reduction of dimensionality is still identified as similar after the transformation [122].
A straightforward method to achieve this is (down) sampling. Instead of using all available data points, only some are selected. By choosing a rate m/n, with m being the length of the original time series and n being the dimensionality after sampling, the dimensionality of a time series can be reduced. For example, choosing a rate of 10, only every 10^{th} data point would be used. For high rates, however, this can significantly influence the shape of the data and therefore disturb the result of an analysis.

A more advanced method is a piecewise aggregate approximation. Similarly to sampling, a rate is chosen. This rate, however, does not just represent the number of data points that are selected from the original time series, but rather several segments that the time series is split into. For each of those segments, then, the data is aggregated using functions such as average [44]. This average value is considered to be representative of the data within each period. Similarly, data within the segments can be represented using interpolation functions. In the case of piecewise linear representation, this function is a linear function of the form $f(x) = kx + d$. For each segment, the coefficients k and d need to be chosen. This can be done by simply connecting the first data point with the last data point for each segment [99]. A more sophisticated method is regression, were the functions coefficients are varied so that for each segment the overall difference between data and the resulting function becomes minimal [159]. Higher order polynomials can be chosen as well, which can reduce the error, but also increases the dimensionality of the transformed time series.

Apart from approximating a time series with a set of lines, other approaches try to improve the downsampling approach by selecting specific points due to defined criteria. In those approaches, the most critical points, for example, certain local minimum and maximum values are selected. Other criteria for selecting those points can be chosen as well.

Another way to represent time series is to convert them to a symbolic representation. Based on segments (intervals) within the time series, those segments are classified according to some label catalogue. This can but does not need to be a subsequent step after piecewise aggregation. In the case of piecewise aggregation, however, the distribution space of the average values can be divided, and a symbol can be assigned to each of the resulting subspaces. Based on this assignment, each segment can be assigned a symbol accordingly, which reduces the dimensionality of the data further [120]. Alternatively, the shape of segments can be described using terms such as "highly increasing transition" [6] or "upward" [145].

Finally, several approaches exist to represent time series, not in the time domain but transform them to another description domain. A widely used method in this regard is the discrete Fourier transformation which was first introduced by Agrawal et al. [4]. Through discrete Fourier transformation, time series data is transformed from the time to the frequency domain. The dimensionality of the time series can be reduced by considering only the lowest frequencies of the result. Several other transformation-based methods such as wavelet transformation or principal component analysis exist.

No matter what kind of representation is chosen, they are a prerequisite for calculation of similarity between time series data.

Computation of similarity between time series is an essential step in the analysis of time series data and often a preceding step for classification and clustering of time series data (see Section 3.1). According to Esling and Agon [62], a similarity measure is defined as a function between two time series which returns a non-negative distance d which indicates how similar the given time series are to each other. It is a fundamental measure of several tasks concerned with the analysis of time series data.

Similarity measures can be computed in two ways. Whole sequence matching uses the whole length of all time series within the period that the search is targeted at. In the case of subsequence matching, on the other hand, a given period is segmented into sub-periods that match a time series according to some measure. This, however, introduces the task of correctly segmenting a larger period. The trivial approach to this, which is fixed length segmentation, however, often is an over-simplification as patterns might appear with different lengths throughout time series. Consequently, the correct choice of window (subperiod) lengths is a non-trivial task that determines the quality of a search result [66].

2.2 Databases and Database Management Systems

In order to be able to do data analysis, large amounts of data are necessary. Those large amounts of data are what is typically referred to as a database. Software that is used to manage this data is called a Database Management System (DBMS). Such systems are built to mediate between (human) agents that want to interact with data and the physical device that is used to store it. When databases were first introduced, the notion that physical implementation and the human-readable modelling of the data could be separated from each other was revolutionary. It led to a significant reduction of implementation effort for software applications.

2.2.1 Relational Model

Before a DBMS can be used to administer data, the respective problem domain has to be modelled. Modelling, in this context, means the abstraction of a real-world situation so that it can be computationally processed.

As a running example, consider a fictional, international network of companies that is comprised of several different manufacturers. Those manufacturers are distributed at different places in Europe and jointly manufacture a product. In order to do so, products need to be sent from one manufacturer to another. Furthermore, different types of sensors can be deployed to each machine. It is possible to assign new sensors to existing machines at any given point in time. Each type of sensor is dedicated to measuring a different physical property and has a sampling rate that determines how many measurements are generated per time unit. This data can be used to monitor the manufacturing

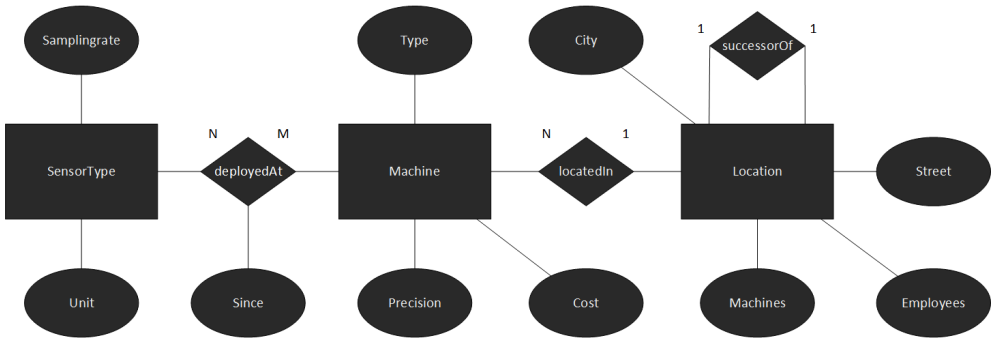


Figure 2.1 Example ERM representing the described scenario.

processes that take place. In order to keep track of the production facilities, their logistical connections and the sensors that are deployed, a database should be created.

First, critical elements and their relations are identified. The result of the modelling process can be visualised as an Entity- Relationship- Model (ERM). A possible visual representation of such a model in the case of the described scenario can be seen in Figure 2.1. Note that many different notations exist to depict this model. In this thesis, Chen-notation [45] will be used.

In the ERM, a schema is defined through entities (depicted as rectangles), relationships (depicted as diamonds) and attributes (depicted as ellipses). Entities are the essential elements and represent things in the real world that exist independently (physically or conceptually). In the presented example, those entities are LOCATION, MACHINE, and SENSORTYPE. Each entity has attributes that describe it. Examples might be CITY or SAMPLINGRATE.

Each attribute can be assigned a value from a defined domain (restricting data types). The cardinality of each relation is denoted by assigning 1, M or N to the respective edge. These numbers represent the maximum cardinality, which is the number of relationship instances that an entity can participate in. The cardinality between SENSORTYPE and MACHINE is N:M (a SENSORTYPE can be deployed to M MACHINE, and a MACHINE can have N SENSORTYPE deployed to it) whereas it is 1:1 for the LOCATEDIN relationship between MACHINES and LOCATIONS (each MACHINE is always in exactly one LOCATION, but every LOCATION can have more than one MACHINE assigned to it). Also, attributes can be added to relationships, as is the case for the DEPLOYEDAT relationship.

Such a model imposes restrictions on data. Data that comply with those restrictions can be put in tables that correspond with the model. Instances of such data can be seen in Table 2.1. As can be seen, not only entities have corresponding tables, but also relations can require a table. DEPLOYMENTS is one such table. It is required due to the cardinality of the corresponding relationship.

2.2.2 Queries over Relational Databases

Queries over databases are used to access the stored data. In order to formulate such queries, many query languages exist. Different paradigms exist to formulate such query languages. Probably the

Table 2.1 Example database tables

Deployment

ID	Machine	Sensor	Since
D1	M1	S2	01.01.2018
D2	M1	S1	25.03.2018
D3	M3	S1	13.08.2018
D4	M2	S3	26.11.2018
D5	M2	S1	20.09.1988
D6	M2	S2	01.01.2018

Sensortype

ID	Quantity	Sampling Rate	Unit
S1	Temperature	20	Kelvin
S2	Power	50	Watt
S3	Force	5000	Newton
S4	Current	4300	Ampere
S5	Luminous Intensity	200	Candela
S6	Position	100	Meters

Location

ID	City	Street	Machines	Successor	Employees
L1	Vienna	Getreidemarkt	M1	Bolzano	7
L2	Bolzano	Dominikanerplatz	M3	London	2
L3	London	Malet Street	M2	Paris	15

Machine

ID	Type	Precision	Workspace	Location
M1	Milling	1	500	L1
M2	Grinding	0.2	50	L3
M3	Turning	1.5	100	L2
M4	Turning	1.1	50	L1

most popular query language is SQL. Some example queries in the case of the presented example database are:

1. "What type of machine is located in Vienna?"
2. "What is the precision of machines that are located in London?"
3. "Which sensor type was last deployed?"
4. "What kind of sensors are deployed to milling *or* turning machines?"
5. "Which are the machines that are *not* turning machines?"
6. "What is the shortest connection between Vienna and Bolzano?"

Queries can be classified based on the requirements they have towards the underlying query language. The most simple queries are conjunctive queries. Queries 1-4 are examples for this class, which in practice makes up the vast majority of queries. They can be extended with union and negation which allow for queries such as 5 and 6, respectively. A final aspect that can be added is recursion. It enables us to ask things like query 6.

In the case of SQL, queries are built from select-from-where clauses. A SQL translation of query 1 would, therefore, have the form shown in Listing 2.1. The results for all three following queries can be found in Table 2.2.

Table 2.2 Results for the presented example queries

Query 1	Query 2		Query 3	
Machines	**Locations.Machines**	**Machines.Precision**	**Sensor**	**Since**
Milling	Grinding	1.5	Temperature	26.11.2018

```
01 | SELECT Machines FROM Location WHERE City='Vienna'
```

Listing 2.1 SQL translation of query 1

Sometimes it might be necessary to combine data that is stored in different tables. In this case, JOIN- statements can be used. In order to answer query 2, for example, data from the SENSOR table has to be combined with data from the LOCATION table via the deployment relationship and the MACHINE table, as shown in Listing 2.2.

```
01 | SELECT Location.Machine, Machine.Precision
02 | FROM Locations
03 | JOIN  Machines ON Locations.Machine=Machines.Type
04 | WHERE Locations.City='London'
```

Listing 2.2 SQL translation of query 2

It is possible to modify these basic blocks further. The DISTINCT keyword can, for example, be used to remove duplicates from a query answer. Also, the results can be ordered in an ascending or descending order for any of the columns in the answer table. Also, the number of results can be limited. In the case of query 3, this can, for example, be used as illustrated in Listing 2.3.

```
01 | SELECT Sensor, Since FROM Deployment
02 | ORDER BY Since DESC
03 | LIMIT 1
```

Listing 2.3 SQL translation of query 3

2.3 Knowledge Representation

The preceding section has covered foundations with respect to data modelling and access in the case of RDBMS, which are widely used in industrial solutions. Other data formats such as CSV or Extensible Markup Language (XML) can be easily transferred to this form. Moreover, while the vast majority of data in the industry is stored in this form, there are some limitations to this approach. From the user's perspective, accessing data can be problematic. It is not only necessary to be aware of (at least some)

of the algebraic properties of the applied query languages, but also the database schema (including the semantic meaning of tables, relations, and attributes within it) must be known in order to be able to formulate queries. This, in reality, leads to the situation that data access by users from outside the computer science domain happens through software applications tailored to specific processes with predefined queries.

As will be shown in Section 3.1, data analysis applications are getting much attention from researchers. This leads to an increasing demand for flexible solutions for the data access problem that are targeted at domain experts from different areas. Such experts, typically, do not have the necessary skills and knowledge to interact with such databases. Even if some experts from one domain by chance can interact with databases, the semantic meaning of the stored data might differ depending on the profession of the respective user. In the presented example, a person responsible for production planning might be just as interested in the stored data as another person concerned with energy management. Their specific interpretation of the same data, however, will fundamentally differ.

Another potential issue of RDBMS is that a fundamental assumption that is made by such systems, in reality, might not hold: Anything that is not explicitly recorded in the database is assumed to be false, which is also referred to as the closed world assumption. For example, in the presented database, one would assume that there is no manufacturer in Berlin or that there is no direct logistical connection between London and Vienna. This, however, in reality often is not the case. In general, we cannot assume that the data we have is complete. In reality, however, it is possible that a manufacturer or a connection exists, but that it is not yet within the database. To be able to consider this, it is necessary to divert from the so-called "closed world assumption" which states that anything not explicitly found within the data is false. Alternatively, the "open world assumption" states that anything that is not explicitly stated to be false is true.

Lastly, and practically most importantly, knowledge can be used to infer additional answers to queries stated by users. For example, if a user asks the presented example database for anything that is a DEVICE, he or she would not receive a result. If however knowledge of the form "machines are devices" were available to the system that answers the query, it would return anything that is stored as MACHINE in the database. To overcome these limitations, it is necessary to be able to represent knowledge along with data and make it available for automated systems.

2.3.1 Logics for Knowledge Representation

Human knowledge is required to make data useful. Already in the previous section, the presented modelling techniques can be seen as a way to formalise this knowledge and make it accessible for software. Those techniques are based mainly on first-order logic (FOL)

First- Order- Predicate Logic

FOL, just as any other kind of logic, is a way to reason. Logic provides a way to convert linguistic sentences into mathematical formulae. Reasoning is the process of coming to conclusions based on some set of premises. Such premises, as well as their conclusions, are formulated as sentences. An interpretation can assign truth values (true or false) to sentences. A set of sentences is called an argument if one of those sentences is the conclusion of the other sentences. In principle, such sentences can have any content. In order to result in a valid argument, however, the sentence that is the conclusion cannot be false, if all the premises are true.

For illustrative purposes, consider the following three example sentences:

1. The factory is in operation.
2. If the factory is in operation, it makes profit.
3. The factory does make profit.

If sentences 1 and 2 are interpreted as premises, and sentence 3 as the conclusion, this would be a valid argument. The third sentence is a "consequence" of the first two sentences.

The presented example uses only propositions, and reasoning on arguments of that sort is investigated in propositional logic. However, there are some limitations to propositional logic. Consider, for example, the following argument:

1. All mechanical engineers wear chequered shirts.
2. Christina is a mechanical engineer.
3. Christina wears a chequered shirt.

We would now like to be able to use sentences 1 and 2 to come to sentence 3 as the conclusion. There is, however, no way to express the connection between sentence 1 and 2 using only propositions. Therefore, predicates are needed. Predicates are properties that can be assigned to objects. The term "first-order" in FOL indicates that variables within predicates must not refer to predicates themselves, which results in also limiting the complexity of reasoning tasks that can be formulated with a given logic. To capture this, formal languages are used. Such languages consist of three elements: an alphabet, terms and well-formed formulae.

A languages alphabet will typically contain constants $(c_1, c_2, \cdots, c_n)$, variables $(x_1, x_2, \cdots, x_n)$, function letters (f_i^n), predicate letters (P_i^n), logical connectives $(\neg, \wedge, \vee, \Rightarrow, \Longleftrightarrow)$, quantifiers $(\forall, \exists)$ and punctuation symbols. For functions and predicate letters, n stands for the arity of the function/predicate, i.e. the number of arguments that the function/predicate takes. The meaning of each of these symbols is given in Table 2.3.

Terms are used to represent objects. In their purest form, constants and variable are terms, but they can also be formed from functions. Consider, for example, the assertion "Meike's mother likes

Table 2.3 Alphabet in FOL

Name	**Syntax**	**Semantics**
Constants	$c_1, c_2, \cdots, c_n$	Abstract representation of concrete objects
Variables	$x_1, x_2, \cdots, x_n$	Abstract representation of generic objects
Functions	f_i^n	Used to build terms from variables or constants
Predicates	P_i^n	Are properties that can be assigned to objects
Negation	$\neg$	Not
And	$\wedge$	And
Or	$\vee$	Or
Conditional	$\Rightarrow$	if ... then ...
Bi- Conditional	$\Longleftrightarrow$	if and only if, (if ... then ... in both directions)
Existential Quantifier	$\exists$	There exists, For some
Universal Quantifier	$\forall$	For all

newspapers". It is not possible to express this correctly using predicates. To express, that there is a particular individual, which is Meikes's mother, and that this individual likes newspapers, functions are used.

Finally, well-formed formulae indicate, that there might also be non-well-formed formulae. To distinguish them from each other, rules exist that define which strings of symbols can be considered in a language. For example, the string "asdfjipabgr" would not be considered well-formed in German. Surprisingly, however, the string "Xylophonspielersessel" would. In FOL, these rules are the following:

- $P_i^n(t_1, t_2, \cdots, t_n)$ is a well-formed formula, where t_i are terms.
- If A and B are well-formed formulae, so are $(\neg A)$, $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$ and $(A \iff B)$.
- If A is a well-formed formula, so is $(\forall x_i)A$.
- If A is a well-formed formula, so is $(\exists x_i)A$.
- Nothing else is a well-formed formula.

So, if we use $P_1^1(x)$ as a predicate representing "... wear chequered shirts" and $P_2^1(x)$ as a predicate representing "... is a mechanical engineer" and the constant c_1 for "Christina" respectively, using the constructs and rules that form the language of predicate logic, the initially stated sentences can be written in the following way.

1. $\forall x P_2^1(x) \Rightarrow P_1^1(x)$
2. $P_2^1(c_1)$
3. $P_1^1(c_1)$

To be able to use FOL for reasoning about semantics (meaning) and answer questions concerned with truth, a way to connect well-formed formulae with objects is required. This is done through FOL-interpretations. An interpretation consists of a non-empty set of objects ($\delta^{\mathcal{I}}$) and the interpretation function, which associates terms of a language with objects in a domain and well-formed formulae with relations over the domain.

FOL is a powerful tool to reason over many real-world problems. As such, it is the basis for Description Logic (DL)s, which in turn will be described in the following section.

Description Logics

In the past section, predicate logic was introduced. It is a way to represent knowledge in an expressive, unambiguous way. Reasoning, in this case, becomes a task that can be carried out algorithmically and consequently is a task that can (in principle) be done by computers. Through logic, it is possible to build "intelligent" software applications that are applied in different fields. Computational complexity is determined by the expressivity of the underlying logic and that practical solutions need to limit this complexity.

Research on this topic led to the field of DLs. These typically are fragments of FOL (which reduces the complexity of reasoning tasks), but might also use constructs that are not captured by FOL. Description logics use a different syntax then predicate logic to express knowledge. In DLs, a domain of interest is abstractly described using concepts and role names. Concepts in this respect are unary predicates (i.e., take one argument) and represent a subset of elements within the domain under consideration. They are built using concept names, role names, and constructors provided by the respective logic fragment. Role names represent binary relations (i.e., take two arguments) between elements.

Reasoning covers several different tasks such as checking concept satisfiability, subsumption, equivalence or disjointness in respect to a given set of knowledge expressions as well as instance or consistency checking, instance classification and query answering. Depending on the expressive power of a given DLs, these tasks can be more or less computationally complex. In the worst case, they can become undecidable (i.e., they can not be completed in finite time). Often only such DLs that allow for tractable formalisms are considered to be practically useful. In DLs, knowledge is organised in two parts, which are defined as follows:

1. TBox: The TBox contains sentences that are concerned with general, terminological (thus the name) knowledge from a given domain. This could, for example, be "Every machine has a location."

2. ABox: The ABox contains assertions about individuals within the domain of interest. An example ABox assertion could be "Meike is a researcher."

Even though this distinction is not significant from a theoretical perspective (all DL statements can be translated to equivalent FOL sentences), it can sometimes be practically useful to distinguish them.

This is especially true in the case of OBDA, as will be shown in section 3.2. Both ABox and TBox together are called a Knowledge Base (KB). In the following section, the basic notion of these words will be illustrated through the use of an example KB.

Any description logic has a specific concept language that defines how concept and role descriptions can be built. Often, C and D are used to denote general concepts, and A denotes atomic concepts (Atomic concepts are concepts that are not themselves defined by other concepts or roles). r and s denote roles. n denotes any natural number. Typically, the semantic meaning of the symbols would be described using model-theoretic characterisation. Even though this interprets the symbols exactly, it also poses a significant barrier for someone who is not used to this notation. As the scope of this thesis is the application of those ideas in the manufacturing domain, despite the apparent problem of ambiguity, the following symbol explanation will, therefore, be given in natural language, followed by some illustrative examples.

1. Conjunction ($C \sqcap D$), also called intersection, denotes anything that belongs to both "C and D".
2. Disjunction ($C \sqcup D$), also called union, denotes anything that belongs to either "C or D or both".
3. The Top- Symbol ($\top$) denotes "anything", or "all". Any element within the domain of discourse is part of this set.
4. The Bottom- Symbol ($\bot$) stands for the empty set or "nothing", which, by definition contains no elements.
5. Negation ($\neg C$) read as "Not C", indicates the complement of concept C.
6. Existential restrictions $\exists r.C$ "exists", represents a set of individuals that have at least one r-successor of type C.
7. Universal restrictions $\forall r.C$ "for all", represents the sets to those individuals that have all r-successors of type C. This is also verified if there are no r-successors.
8. Number restrictions $\leq nr.C$ and $\geq nr.C$ are used to define the maximum or the minimum number of relations that an individual has. These can both be qualified (with concept C), or unqualified (without concept C). This is equivalent with $\exists r.C$ for $n = 1$
9. Nominals $\{a\}$ can also be used to define concepts as a set of objects that belong to this concept.

The aforementioned constructors can be used to create sentences by combining concepts or roles through inclusion statements of the form $C \sqsubseteq D$ ("C is subsumed by D"). More expressively, this means that any individual that is within C is also within D. Similarly, equivalence $C \equiv D$ can be used to define that two concepts are logically interchangeable.

Just as constructs for concepts exist, also relations between roles can be defined.

1. Role inclusions $r \sqsubseteq s$ are used to indicate that some roles are specializations of other rules. If several roles are used ($r_1, r_2 \cdots \circ r_n \sqsubseteq s$), these are referred to as "complex role inclusions"
2. Role disjointness $Disjoint(r,s)$ is used to express that no two individuals can have both roles r and s.

Example Knowledge Base

To illustrate the aforementioned notions, we return to the running example introduced in section 2.2. Using the constructors introduced above, it is possible to formulate statements regarding relevant domain knowledge.

Given the atomic concepts MACHINE and SENSOR, using the intersection, for example, complex concepts like SENSORYRIG could be defined as any machine that is also a sensor. This could, for example, be used to talk about machines that also generate monitoring data without external sensors.

$$SensoryRig \equiv Sensor \sqcap Machine$$

Similarly, the union can be used to define a superclass for both SENSORS and MACHINE. Possibly, we would like to refer to objects from either of these groups as DEVICES. Such a concept can be defined in the following way:

$$Device \equiv Sensor \sqcup Machine$$

Using the Top- Concept, it is furthermore possible to say that, within our domain of interest, anything is either a DEVICE or a LOCATION or both.

$$\top \sqsubseteq Location \sqcup Device$$

Similarly, we could define that nothing is both a DEVICE and a LOCATION

$$Location \sqcap Device \sqsubseteq \bot$$

Negation can be used to express, for example, that any LOCATION that is not also a SUCCESSOR is the SOURCE of the supply chain.

$$Location \sqcap \neg Successor \sqsubseteq Source$$

Existential restrictions can be used to define concepts by using roles (and vice versa). For example, a SENSOR could be defined as anything that is used to measure any other instance.

$$Sensor \equiv \exists measures.\top$$

Universal restrictions, on the other hand, can be used to express that MILLINGMACHINES are those MACHINES that are deployed to MILLINGLOCATIONS.

$$MillingMachine \sqsubseteq \forall deployedTo.MillingLocation$$

Through number restrictions, the concept RESEARCHMACHINE could be defined as any MACHINE that has at least five SENSORS deployed to it.

$$ResearchMachine \sqsubseteq \geq 5 deployedTo.Machine$$

To define our PRODUCTIONNETWORK, nominals can be used.

$$ProductionNetwork \equiv \{Vienna, Bolzano, London, Paris\}$$

Role inclusions could furthermore be used to define that anything that the role CONTROLS is a combination of the roles MEASURES and MANIPULATES.

$$controls \sqsubseteq measures \circ manipulates$$

Given this simple KB, we could now start to reason over the domain. Given an instance that is declared to be a MACHINE, we could, for example, conclude that it is not a LOCATION. Also, we would not expect it to be a SOURCE of the supply chain. Reasoning, however, can also lead to unwanted results, if the axioms are not defined correctly. Correct, in this respect, however, can change over time. We could, for example, introduce an instance "Judith" which represents a researcher. A corresponding concept RESEARCHER was not used until now. It would be reasonable to define this not to be a Device. Through reasoning, however, our system would conclude that "Judith" must be a LOCATION. If we rule this out as well, we will end up with an inconsistent KB. In this example, this is due to the axiom $\top \sqsubseteq Location \sqcup Device$, which states that anything is either a LOCATION or a DEVICE. Inconsistencies like this can occur due to implicitly formulated requirements that result from the statements which is why care needs to be taken when defining such KBs. In the example case, the existing axioms would need to be adapted accordingly.

2.3.2 Semantic Web Technologies

In the previous section, DLs were introduced. They are closely linked to ontologies, which had their arguably most prominent use in the course of Tim Berners-Lee's vision of a Semantic Web, which he proposed as an evolution of the existing world wide web by annotating web content with semantic meaning in order to make it more accessible to both humans and computers [23]. Ontologies, in this vision, would be the tool of choice to define said meaning. DLs have formal, logic-based semantics and existing decision procedures tailored for implementation in automated reasoning systems. Therefore, they were identified as a suitable choice for languages that are used for this task. This resulted in the

```
01 | <Bolzano> <a> <City>.
```

Listing 2.4 Example RDF triple

development and later the standardisation of Web Ontology Language (OWL) by the World Wide Web Consortium (W3C). These standards are important for this thesis, as they will be used to implement ontologies for the systems developed in Chapters 4 and Chapter 5.

Semantic Web Stack

To make the vision of a semantically annotated world wide web reality, the W3C developed several tools. Those tools and their relations with each other are depicted in Figure 2.2. In the scope of this thesis, the most relevant parts of this stack are Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), OWL and the SPARQL Protocol And RDF Query Language (SPARQL).

RDF is the basis of the semantic web. RDF statements express relationships between two objects. The relationship is directionally and is called a property in RDF. The related objects are connected via a *predicate* and are called *subject* and *object* respectively. Because RDF statements consist of three elements, they are called triples [147]. RDF is a graph-based model allowing for linking data from heterogeneous sources. Syntactically, the W3C standard solution for storing such graphs is to use the XML format. Other formats such as Turtle or JSON-LD exist. Data is expressed as a list of statements, which follow a simple triple schema. For example, the statement "Bolzano is a city" could be expressed as shown in listing 2.4, where "Bolzano" is the subject, "a" is a predicate and "City" is an object.

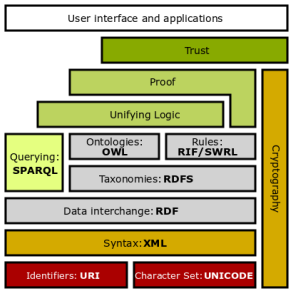


Figure 2.2 Semantic Web Stack [23]

RDFS provides basic data-modelling functionality for RDF data. Using a predefined set of classes and properties, simple semantics such as groups and relationships between resources can be formulated. These include property domains and ranges as well as subconcept ("is a") relationships between concepts.

The vocabulary provided by RDFS is very limited. In order to be able to express more complex parts of human knowledge, a more expressive language is required. This is the role of OWL. OWL is built on the foundations of description logics and allows knowledge representation in a machine-

readable way. Different fragments of OWL exist that correspond to DLs of varying expressivity and consequently varying complexity of reasoning tasks.

Finally, in order to access and manipulate data formulated in RDF via ontologies formulated in RDFS or OWL, a query language is needed. For this, the W3C recommends SPARQL. Using the concepts and properties of ontologies, graph patterns are built. These graph patterns can be seen as filters. They are compared with the data graph, and any graphs matching the specified patterns are returned. Results of such queries can either be graphs, data in a tabular form or an answer to a specific question.

Apart from the Semantic Web, ontologies are useful for other fields of application such as conceptual modelling, information, and data integration and OBDA. In the case of OBDA, the goal is to provide easier access to large amounts of data through a high-level, conceptual view. The corresponding TBox can then be seen as the explicit representation of this view. Together, ABox and TBox can then be seen as a KB which can be subject to reasoning tasks. Typically, those tasks are answering conjunctive queries to the ABox while taking into account the knowledge expressed in the TBox. These queries can be encoded into SQL- queries as introduced in Section 2.2.2. In order for OBDA systems to be feasible from a computational perspective, however, the expressivity of the used languages needs to be limited. This leads to $DL-Lite$.

DL-Lite

Expressive DLs (i.e., providing more freedom when defining classes and roles regarding constructors) increase the complexity of reasoning tasks, potentially to an extent where they become undecidable. However, even for decidable logics, complexity can become a severe problem limiting the applicability of systems building on it. This is especially problematic in the presence of large ABoxes which are, for example, prevalent in many applications such as data integration and the knowledge graphs. The expressive power of the current standard OWL 2 might not be necessary or even infeasible due to the size of a given KB, and therefore it is useful to trade some expressive power for complexity reduction. In this case, three different "profiles" exist, one of which being OWL 2 QL which was specially tailored to deal with vast amounts of data and query answering. This profile is therefore used by OBDA- applications.

By complexity, more specifically computational complexity, unless explicitly specified differently, worst-case complexity is meant. According to Dean, complexity is defined in the following way: "*In computational complexity theory, a problem X is considered to be complex in proportion to the difficulty of carrying out the most efficient algorithm by which it may be decided. Similarly, one problem X is understood to be more complex than another problem Y if Y possesses a more efficient decision algorithm than the most efficient algorithm for deciding X*" [57]. Complexity is a measure for how much operations are needed to computationally complete a specific (reasoning) task for a given input. Complexity classes can be defined to be the set of problems for which there exists a decision

procedure with a given running time or running space (memory) complexity. These are measured by the size of the input (n).

To address the trade-off between expressivity and complexity, Calvanese et al. [37] introduced $DL-Lite$, which is a family of description logics. $DL-Lite_{\mathcal{R}}$ was later chosen to be at the basis for the OWL 2 QL language. They present algorithms that can be used to answer unions of conjunctive queries and show that the complexity is LogSpace (using a logarithmic amount of writeable space) in the size of the ABox, which is well below what is considered the threshold for practically applicable solutions. A notable consequence of their findings is that their queries can be rewritten to equivalent SQL- queries. The $DL-Lite$ family is built around $DL-Lite_{core}$. In this language, a TBox is formed by a finite set of inclusion assertions of the following form.

$$B \sqsubseteq C$$

B denotes a basic concept (a concept that can be an atomic concept or a concept of the form $\exists R.\top$). R denotes a basic role (a role that is an atomic role or an inverse of an atomic role). General concepts (C and D), can be built from basic concepts or their negations. Similarly, general roles are defined as roles that are built from basic roles or their negations. Furthermore, general concepts are only allowed on the left-hand side of inclusions. Finally, on the right-hand-side cannot be a union or the negation of a complex concept.

ABoxes are built using $A(a)$ to assert that the object denoted by the constant a is a member of the atomic concept A. This can be expanded to assertions of the form $C(a)$. Similarly, $P(a,b)$ can be used to assert roles between objects denoted by a and b.

Based on $DL-Lite_{core}$, the other members of the $DL-Lite$- family are defined, one of which being $DL-Lite_{\mathcal{R}}$. Specifically, this language adds to the expressions allowed in $DL-Lite_{core}$ qualified existential quantifications of concepts in the form of $\exists R.C$ Furthermore, it is possible to assert role inclusions of the following form, with E denoting a general role (similar to general concepts C).

$$R \sqsubseteq E$$

2.4 Ontology-Based Data Access over Temporal Data

Until now both databases and knowledge representation were discussed. For the scope of this thesis, however, their combination in the form of OBDA is at the focus. Therefore, the fundamental principles of OBDA will be described in the following section. Furthermore, the possible extensions of this technology which can be used to accommodate requirements due to the presence of temporal data will be described.

2.4.1 Ontology-Based Data Access

As already mentioned, databases today are the standard solution for storing data. They do however have some properties that limit their applicability. Those are:

- **Application Focus:** Databases are built to interact with (software) applications. Direct interaction with human users requires the latter to know not only the respective language (often SQL) but also the schema of the database. Schemas, however, can have cryptic column and table name identifiers and store redundant data which often is not required by domain experts.
- **Data Integration:** A common problem in manufacturing engineering applications is the integration of different autonomous systems into larger ones that facilitate features of the subsystems. In the case of databases, data that is stored in different systems has to be combined, which is referred to as data integration. This, however, is a challenging task, especially for large and complex systems.
- **Incompleteness:** Data stored in databases is incomplete. Often, domain-specific knowledge is needed to infer facts that are needed to use the data. As this knowledge is not explicitly stored along with the data, databases are not as useful as they could be for people other than those that initially created the respective database.

A promising way to overcome, at least parts of these limitations, is OBDA, which was first introduced by Poggi et al. [144]. Several research papers were published on the topic as well as the first industrial implementations presented. More details on some of these publications are given in Chapter 3.

The main idea behind OBDA is to use ontologies as an interface for users that allows them to access data stored in (potentially several) data sources abstracting the details on the data layer. It is important to note that in this context data sources are considered to be conventional (i.e. relational) database management systems. Furthermore, data does not need to be moved or even changed physically. No materialisation is necessary, which is why this technology also is referred to as virtual knowledge graphs. An OBDA-instance, according to Xiao et al. [179], consists of the pair $(\mathcal{P}, \mathcal{D})$. In this pair, $\mathcal{P}$ depicts an OBDA specification and $\mathcal{D}$ the source database. The OBDA specification $\mathcal{P}$ itself has three independently existing components: $\mathcal{O}$ (Ontology), $\mathcal{M}$ (Mapping) and $\mathcal{S}$ (Data source schema).

The ontology represents domain knowledge and serves as the interface between user and data. It exists autonomously from the data layer, and therefore the systems can be maintained separately. The mapping $\mathcal{M}$ connects concepts and roles from within the ontology with the data source. Lastly, the database schema $\mathcal{S}$ imposes restrictions on the data as described already in Section 2.2.1.

To illustrate this further, consider the following ontology, which could be used to access a manufacturing related database such as the one used in Section 5.3:

$$MillingMachine \sqsubseteq Machine \tag{2.2a}$$
$$MillingMachine \sqcap GrindingMachine \sqsubseteq \bot \tag{2.2b}$$
$$HighPrecisionMachine \sqsubseteq Machine \tag{2.2c}$$
$$\exists hasSensorDeployment.\top \sqsubseteq Machine \tag{2.2d}$$
$$\exists hasSensorDeployment^{-}.\top \sqsubseteq Sensor \tag{2.2e}$$

These assertions describe that every MILLINGMACHINE also is a machine (Equation 2.2a), that nothing can be both a MILLINGMACHINE and a GRINDINGMACHINE (Equation 2.2b). Furthermore, there exist machines which are considered to be HIGHPRECISIONMACHINE (Equation 2.2c)and anything that has an outgoing HASSENSORDEPLOYMENT relation is a machine (Equation 2.2d). Incoming HASSENSORDEPLOYMENT relations indicate that the respective entity is a SENSOR (Equation 2.2e).

Reusing the example database presented in Section 2.2, this ontology could be used to access data stored. Before that, however, the ontological concepts and relations need to be connected to the data instances stored in the source database. These connections are established through mapping assertions. Generally, mapping assertions have the form $\phi(x) \rightsquigarrow \psi(x)$. Here, x denotes queries and $\phi(x)$ denotes formulas, which, for the scope of this thesis can be represented by SQL-queries. $\psi(x)$ denotes ontological concepts and relations from the ontology $\mathcal{O}$.

For the given example, such a mapping could have the following form:

```
SELECT ID FROM Machine WHERE Type='Milling'⇝MillingMachine("Machine/"+ID)
```

Another potentially useful mapping could be used to populate the HIGHPRECISIONMACHINE concept. This allows accessing all machines that are considered to fulfil the requirements for "high precision" without having to know the exact threshold.

```
SELECT ID FROM Machine WHERE Precision >1⇝
HighPrecisionMachine("Machine/"+ID)
```

In the context of OBDA, query answering can be seen as an inference task. Given a query, the goal of this task would be to return all constants from $\mathcal{D}$ that are a certain answer. Such certain answers can formally be described as $\mathcal{I} \models q(a)$ for every model $\mathcal{I}$ of $(\mathcal{P}, \mathcal{D})$. The combination of conventional databases and ontologies creates a virtual knowledge graph, which can be accessed like any other knowledge graph via SPARQL. Consider, for example, the query shown in Listing 2.5, which returns all certain answers to a query for all entities that are considered to be MACHINES.

```
01 | SELECT ?Machine ?Sensor
02 | WHERE {
03 | ?Machine a :Machine;
04 | :hasSensorDeployment ?Sensor.
05 | }
```

Listing 2.5 Example SPARQL Query

Even though both, the presented query and the source database are reasonably simple, it is evident that this query is much more readable and intuitive than its SQL equivalent. Moreover, to formulate such a query, a user does not need to know anything about the database schema.

Furthermore, consider that the ontological concepts could be mapped to entities in several different, distributed databases. The query would not change, as the user would still only be interested in seeing all MACHINES and SENSORS that are connected with them- no matter where that information is stored.

In practice, an OBDA system is created by domain and IT experts together. Domain experts, together with ontology engineers, would explicitly write down relevant domain knowledge, whereas IT experts define respective mappings to the existing data sources. This might, on the one hand, seem like much effort, but especially in a setting where domain experts regularly need to interact with data, this reduces the demand for time-intensive interactions between domain experts and IT-experts, whenever data needs to be accessed, significantly.

As can be seen, this approach potentially solves the limitations mentioned above of conventional RDBMS. An ontology can be used as an interface to the data for human users. Furthermore, such an ontology can be linked to different data sources which might exist independent from each other. In such a setting, the ontology can be seen as a unifying layer which leads to the integration of heterogeneous data sources. Here, users can interact with almost arbitrarily complex database systems (or combinations of them), without having first to understand their structure. Furthermore, through the use of an ontology, domain-specific terminology can be used, which makes the interaction with data much more natural for domain experts.

Lastly, due to the use of logics, in OBDA, query answering becomes a reasoning task. Consequently, explicitly written down knowledge in the form of ontologies can be used to infer new facts from databases which are not recorded explicitly. In the presented example, the ontological concept MACHINE is defined as the superclass of the concept MILLINGMACHINE. A query for all instances of MACHINE would, in a conventional database, not yield any results. Due to inference, however, an OBDA system would return anything that is mapped to either MILLINGMACHINE or MACHINE, or both. Similarly, the concepts TURNINGMACHINE or GRINDINGMACHINE could be imagined to be useful.

2.4.2 Approaches to Temporal Data

Conventional OBDA approaches do not provide specific tools dedicated to improving access to temporal (i.e. time-stamped) data. This is also reflected by the fact that the standardised ontology languages proposed by the W3C OWL 2 QL and OWL 2 EL were designed to only reflect static knowledge. Recently, however, some approaches to overcome this limitation were presented. This thesis also is concerned with temporal data, which is why some of the fundamental ideas of these extensions will be described in the following sections.

According to Artale et al. [11], three temporal data models can be distinguished:

- **Point-based:** Time is assumed to be discrete, and assertions come with an associated time-point at which it is true.
- **Interval-based:** Assertions are not associated with a time-point, but rather with an interval in which they are true.
- **Dense Time:** Instead of being seen as a flow of discrete values, time is considered to be a dense (i.e. continuous) flow.

Point- Based Approaches

In point-based approaches, facts are reflected in the form of time-stamped ABox assertions. These are collectively referred to as temporal ABox. Similarly to "regular" ABox assertions they can either be concept assertions ($A_k(a_i,n)$) or role assertions ($P_k(a_i,a_j,n)$) where $n \in \mathbb{Z}$ or $n \in \mathbb{N}$ represents a timestamp.

Using this, most languages that were proposed to capture a point-based notion of time are subsets of FOL. Apart from domain variables, just another variable type representing instances from the time domain (time-stamps) is introduced. Consequently, reasoning over point-based ontologies results in reasoning over FOL. Consider, for example, the following formula that might be used to describe that at no point in time a finished product can exist that also is currently processed by a machine.

$$\forall t \forall x \forall y (\text{FinishedProduct}(x,t) \wedge \text{isMachinedby}(x,y,t) \Rightarrow \bot).$$

It is known, that without further restrictions, FOL is undecidable and therefore, useful subsets need to be defined to acquire decidable, or even practically feasible complexity classes for query answering tasks. These approaches can be distinguished in domain-centric and time-centric, depending on the kind of simplifications that are made.

Intuitively, domain-centric approaches restrict temporal quantifiers (i.e. $\forall t$ in the example above), and time-centric approaches restrict quantifiers on domain variables (i.e. $\forall x$, $\forall y$ in the example above). A simplified representation of time-centric approaches is Linear Temporal Logic (LTL) [143]. It introduces constructs representing the temporal relationships "since" and "until", which can be used to describe things like "sometime in the past/future"($\diamond_{P,F}$) or "always in the past/future"($\Box_{P,F}$).

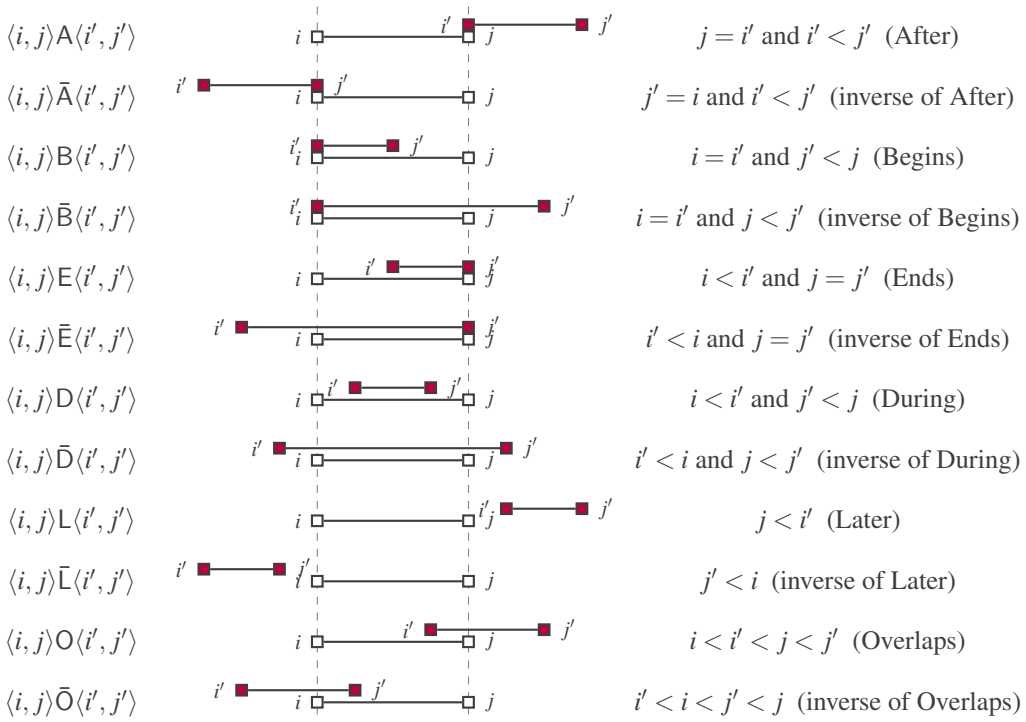


Figure 2.3 Allen's interval relations from [11].

Interval- Based Approaches

The assumptions made by point-based approaches might cause problems when it comes to modelling specific domains. Not only can modelling of facts that refer to intervals instead of points in a point-based approach lead to significant performance issues, but also can it lead to nonsensical interpretations for specific concepts. Consider, for example, the following instance from a database table reflecting entries from the columns (PeriodStart, PeriodStop, MeanValue): (08:00, 08:05, 42). Reflecting this point-wise in the form of, for example, a series of facts of like (08:01, 42), (08:02, 42), ... would not make sense, as the meaning of mean value is either be contradicted or lost in the process.

Consequently, a way to reflect intervals is necessary. A standard way of expressing temporal relationships between intervals is provided in the form of Allen's relations [5] depicted in Figure 2.3. Here, $\langle i, j\rangle$ and $\langle i', j'\rangle$ denote temporal intervals between the time-points i, i' and j, j' respectively.

Using these constructs, it is possible to express things like, for example, that after each power overload, eventually, the tool will break. Note, that in this formula the variables χ and ρ refer to temporal intervals instead of points in time.

$$\forall\rho\forall\chi\forall x(\text{PowerOverload}(x,\rho) \Rightarrow \forall\chi(\chi \text{A}\rho) \Rightarrow \text{ToolBreak}(x,\chi)).$$

Halpern- Shoam interval temporal logic ($\mathcal{HS}$), introduces diamond-like operators for Allen's relations [82]. Intuitively, this makes it possible to use, for example, the diamond version of the "after" ($\langle A \rangle$) operator to express "sometime after".

$\mathcal{HS}$ is undecidable, but recently, some decidable versions of this logic were introduced. For example, a combination of $\mathcal{HS}$ with a simplified version of $DL-Lite$ called $\mathcal{HS}-Lite^{\mathcal{H}}_{horn}$ was introduced. Query answering tasks in this logic are PTime-complete for both combined and data complexity [10].

Lastly, Halpern-Shoam logic can be extended to reflect intervals in more than one dimensions resulting in datalog$\mathcal{HS}^{\square}_{n}$. Using this logic, it is possible to describe, for example, that over some temporal interval another quantity, which can also be interpreted as an interval (for example a power measurement) increases or decreases.

Dense Time Approaches

Until now, time-points n were considered to be elements from a discrete domain ($n \in \mathbb{N}$ or $n \in \mathbb{Z}$). For real-world applications, this can potentially lead to problems. Especially, it metric criteria are to be considered. Such criteria could, for example, be used to define that a concept holds if some criteria hold for a minimum/maximum duration in time (see example below).

In such settings, once the resolution in the temporal domain is fixed, it can not simply be increased without also having to change all concept definitions that use metric temporal arguments. To address this, a solution is to consider time to be dense (continuous), which essentially means that between any two points in time there can potentially be infinitely many more time points. In other words: $n \in \mathbb{Q}$ or $n \in \mathbb{R}$.

A way to capture this was proposed by Brandt et al. [28]. $MTL^{nr}_{datalog}$, which combines datalog rules (another option for ontology languages aside from for example DLs) with Metric Temporal Logic (MTL). Operators such as "for some interval ρ in the future/past" ($\diamondplus_{\rho}$ and $\diamondminus_{\rho}$, respectively) and "for all intervals ρ in the future/past" ($\boxplus_{\rho}$ and $\boxminus_{\rho}$, respectively) can be used to define temporal relationships between concepts.

An illustrative example of how $MTL^{nr}_{datalog}$ can be used to model real-world applications is given by Brandt et al. [29]. In this example, a complex temporal concept PURGINGISOVER, which is required by service engineers concerned with monitoring data coming from gas turbines, is defined. The concept (depicted in Figure 2.4) is defined as follows:

1. Two sensors, a rotor speed sensor and the other one a temperature sensor are deployed;
2. These sensors are co-located in some part of the same turbine;
3. The temperature sensor registers that the main flame was burning for **at least 10 seconds**;
4. within the preceding 10 minutes: rotor speed sensor measures the speed of at most 1000 rpm for **at least 30 seconds**;

5. within preceding 2 minutes: rotor speed measured at least 1260 rpm for **at least 30 seconds**.

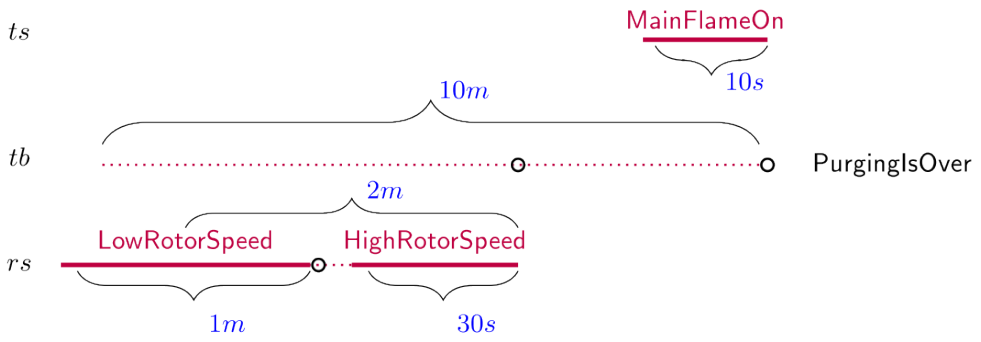


Figure 2.4 Visualisation of the concept PURGINGISOVER from [29]

As can be seen, the last three criteria are temporal. In order to capture them, the following $MTL^{nr}_{datalog}$- rules can be used.

$$\begin{aligned}\text{PurgingIsOver}(tb) \leftarrow & \boxminus_{[0s,10s]}\,\text{MainFlameOn}(ts),\\ & \diamondminus_{[0s,10min]}[\boxminus_{[0s,30s]}\text{HighRotorSpeed}(rs),\\ & \diamondminus_{[0s,2min]}\boxminus_{[0s,1min]}\,\text{LowRotorSpeed}(rs)],\\ & \text{ColocTempRotSensors}(tb,ts,rs).\end{aligned}$$

$$\begin{aligned}\text{HighRotorSpeed}(tb) &\leftarrow \text{rotorSpeed}(tb,v), v > 1260.\\ \text{LowRotorSpeed}(tb) &\leftarrow \text{rotorSpeed}(tb,v), v < 1000.\end{aligned}$$

$$\begin{aligned}\text{ColocTempRotSensors}(tb,ts,rs) \leftarrow & \text{isMonitoredBy}(pt,ts), \text{TemperatureSensor}(ts),\\ & \text{isMonitoredBy}(pt,rs), \text{RotationSpeedSensor}(rs),\\ & \text{isPartOf}(pt,tb), \text{Turbine}(tb).\end{aligned}$$

As can be seen, several different approaches exist to model temporal relationships between concepts. The correct choice for modelling approaches heavily relies on the respective domain of interest and therefore needs careful interaction with the respective domain experts. In the following Chapter 3, however, some more research results on the field of temporal OBDA will be presented.

Chapter 3

Literature Review

In this section, first, in order to identify requirements of the manufacturing domain towards data access software, current research directions on data mining in the manufacturing domain will be surveyed. Then, in order to find promising OBDA approaches, research initiatives in this field and their findings will be observed.

3.1 Data Analysis in Manufacturing and its Limitations

As already mentioned in the introduction chapter, data analysis originally refers to a step in KDD, which was first introduced by Fayyad and Uthurusamy in 1996 [63]. According to them, data mining is defined as "*...applying data analysis and discovery algorithms that, under some acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data*". Data mining is often used synonymously with other terms such as "Artificial Intelligence", "Machine Learning", "Soft Computing" or "Pattern Recognition". To minimise ambiguity, however, those terms will be omitted if possible.

Methods in data mining are primarily based on methods from statistics. Typical tasks in respect to dealing with time series data, according to to Esling and Agon, are the following [62]:

- Query by content: The goal is to retrieve a set of solutions that are similar to what the user has asked for. The argument of such a query itself typically is a time series, which is compared with those that are stored within the database that is subject to the query. An example query from manufacturing research could, for example, be "given this sequence of force measurements, return all similar measurements".

- Clustering: Given data, a finite set of clusters is found. Clusters are groups of data that are homogeneous and distinct to other clusters. Time series data generated by an energy monitoring system could, for example, be clustered in order to find characteristic modes of operation and determine base-load and peak-load levels.

- Classification: Given a set of time series, the goal is to assign labels to each of them. In this respect, classification is similar to clustering, but in the case of classification, the labels are known. Furthermore, there exists a dataset (time series data with their respective labels) that is used to "train" the algorithm on how to assign labels to new (i.e., unlabelled) time series. Classification of time series from the manufacturing domain could, for example, be used to label quality data as OK/not OK based on some (implicit) criteria.

- Segmentation/Summarization: The goal is to reduce the dimensionality of a time series by finding segments within them. If applied successfully, the characteristic features are retained. This can be achieved through piecewise approximation via functions or splines. Given long-term measurements of manufacturing machines, the identification of time series that are representing non-production processes is an example application. This information can, for example, be used to clean data and reduce storage demand.

- Prediction: Under the premise that time series data has some regular behaviour, historical values can be used to predict the development of future measurements. Similar to the classification task, algorithms are trained with historical data. The result is a model of future behaviour that can be used to predict future outcomes. An example is the prediction of energy demand of industrial chillers based on different load scenarios, as it is described in Chapter 4.

- Anomaly Detection: Abnormal segments within time series (anomalies) are often an indicator for process irregularities and are therefore interesting for analysts. Using a model that represents the expected behaviour of measurements, those segments that do not comply with this behaviour can be identified. An example application is the identification of an unstable production process based on acceleration measurement.

Regarding applications of such data analysis techniques in the manufacturing domain, surveys were written by Harding et al. [83] as well as Choudhary et al. [52]. Harding et al. [83] identified categories within which work was done. Within each of these areas, some representative developments are named in order to give an overview of common goals and issues.

3.1.1 Analysis of non-temporal Data

An example for analysis of non-temporal data is given by Rawat and Attia [149], who investigate the influence of process parameters and tool wear on the machinability of such Carbon- Firbe Reinforced Polymer (CFRP). Even though they also analyse time series data such as temperatures or forces in their work, machinability maps such as the one presented in Figure 3.1 can be created without the use of time series data.

Mining data generated by sensors in production processes can be used to gain knowledge about the underlying processes and thereby help to improve them. Especially in the manufacturing domain, dedicated experiments to analyse such processes are often very costly due to high machine and

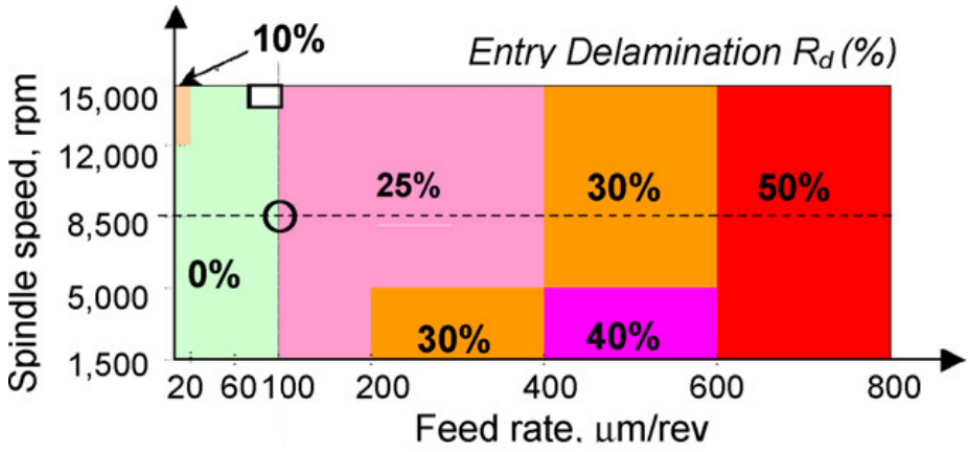


Figure 3.1 Example machinability map from [149] showing the chance for entry delamination at different sets of process parameters.

material costs. Data mining can be used to reduce the number of necessary experiments by efficiently using process data that is generated throughout production processes without having to rely on dedicated experiments entirely. This approach is pursued, for example, by Chien et al. [49, 46]. They apply data mining to analyse process data automatically collected to identify "retrospective design of experiments" that match potential experimental designs formulated by researchers. Specifically, instead of defining experiments and only afterwards collecting the respective data, they mine available data and match this with potential experiments. In their case, the goals of those experiments were to narrow down the number of potential causes for low production yield in semiconductor wafer production. Based on this, the authors went on to propose an approach to enable data analysis for production yield improvement in the semiconductor domain that can cope with the growing problem of smaller production lot sizes that result in a higher frequency of ill-understood production processes [50].

Another instance of decision support is presented by Rodríguez et al. [152]. Intending to build a tool that automatically proposes the best-suited milling tool for a given task, they carry out experiments with a range of tools and use different data mining algorithms to create models that are capable of predicting tool wear and surface roughness. The scope of this work, however, is relatively small. Only four types of tools with very similar features were investigated. This highlights one of the main issues of data mining in the manufacturing domain: specific data is limited for single organisations/ use cases as experiments usually are time-consuming and therefore expensive.

Often, fault detection is a worthy field for analysis of non-temporal data. An overview of this is given by Köksal and Batmaz [106]. Furthermore, several more recent applications concerned with improving product quality and fault detection based on data mining can be found in the literature. Especially the semiconductor industry seems to be an essential field in this respect. Chien et al. [48], for example, investigate a system for live monitoring of wafer faults based on spatial statistics in combination with neural networks.

Chongwatpol [51] describes an industrial use case with three potential approaches to use data mining for fault detection in the manufacturing industry. Kamsu-Foguem et al. [94] illustrate the potentials of association rule mining for quality improvements in production processes. They also highlight the necessary interaction between expert knowledge and knowledge that is discovered through data mining. Not only can (human) domain knowledge be used to evaluate automatically generated knowledge, but also the potential of ontologies to provide additional information that guides data mining is emphasised.

3.1.2 Analysis of Temporal Data

Probably the largest group of temporal data analysis tasks in the manufacturing domain is the prediction of product quality based on process data.Since then, several works emerged concerned with improving milling [121, 183, 182, 31, 7, 58, 152], turning [97, 76, 168, 77], drilling [67, 32, 74]. Most of the authors used time series data such as forces ([121, 7, 58, 77, 32, 74]), accelerations ([183]) or other quantities([182, 152, 97]) for their analysis. Mining of production process data can also be used to extract information regarding the capabilities and constraints of manufacturing systems, as Shahbaz et al. show [158].

In Figure 3.2 data selection that was carried out by Yoon et al. [180] is depicted. In order to be able to estimate models for an empirical power-consumption model based on process parameters for milling, they experimentally determined the power input of a machine in various conditions (tool wear, spindle rotational speed, feed rate, depth of cut). Due to the properties of the investigated machine, transient periods in the beginning and the end of each process were found. Due to the nature of the model they used, only constant power values could be used for model estimation and therefore those transient periods had to be omitted. The data within the resulting measurement period, however, was used to fit parameters of polynomial functions.

Another example of this type of data analysis task is the work of Wang et al. [176]. Based on force measurements from a milling tool, they try to predict the development of tool wear. Based on data that they acquired from dedicated experiments, they evaluated a novel classifier. In total, 400 samples were generated. For each sample, process forces are measured and recorded. In this evaluation, they investigate the importance of different features both from the time and the frequency domain. Features in the time domain included root mean square, variance, peak to peak value, kurtosis and skewness of the raw time series values.

As shown in [177], different measurements can be used to identify tool failure before it will occur. This helps to reduce downtimes and waste. Identification of imminent tool failure is typically a task that is continuously carried out using streaming data. Before this, however, can be implemented, the rules to determine it need to be identified, which is a data analysis task. The idea to anticipate immanent tool breakage based on time series data is also illustrated in Figure 6.14a. There, energy measurements are used to identify tool breakages due to an increased electrical power requirement. This idea will also be used in Section 6.5.2

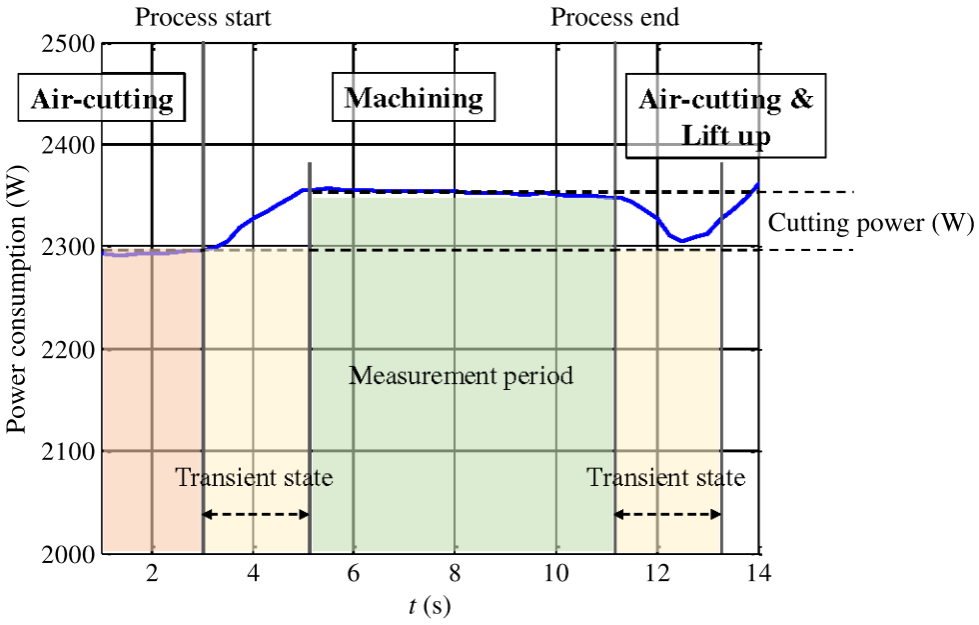


Figure 3.2 Scenario for energy measurement analysis [180].

machine tool	material	tool	v_c in m/min	a_p in mm	a_e in mm	f in mm
HERMLE C42U	2.4668	Walter TAA H2EC38217-E10-10-1	60	5	2	0.2

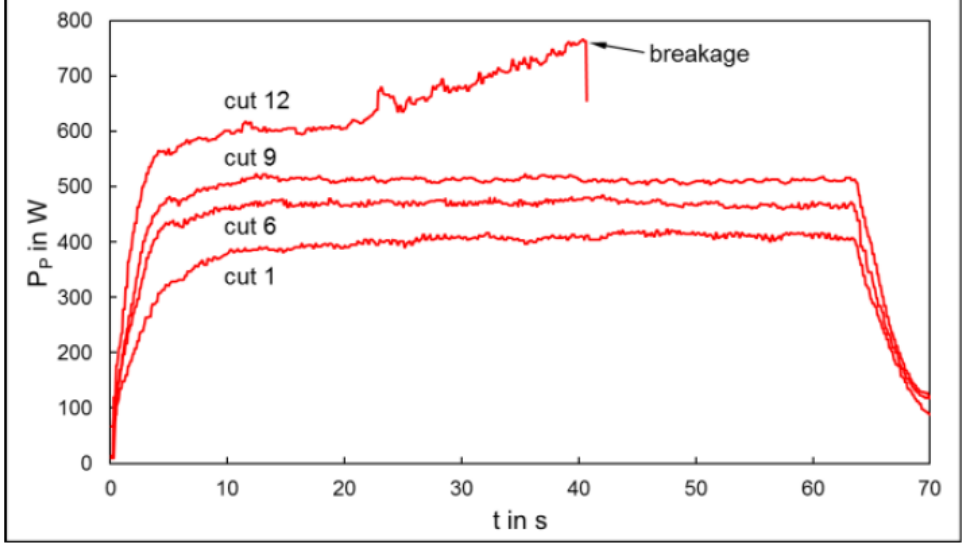


Figure 3.3 Development of power measurements for a series of cuts [81].

Table 3.1 Example events identified by stream reasoning

Reasoning	Event
Average energy use smaller then idle threshold and spindle speed=0	Machine idle
Spike due to spindle startup (0–8000rpm)	Expected energy spike
Spike due to spindle speed increase (8000–16,000rpm)	Expected energy spike
Previous two idle periods energy use constant at 124 kJ	Idle energy constant
Energy spike unaccompanied by shift in spindle RPM.	Anomalous spike
Current idle period energy use (211 kJ) >past idle period average energy use (124 kJ)	Idle energy increase
Current part energy (1218 kJ)> previous parts average energy (1087 kJ)	Part energy higher
Idle energy increasing monotonically over past two periods (342 kJ>211 kJ>124 kJ)	Idle energy trend

Vijayaraghavan and Dornfeld [174] show how power measurements in combination with spindle speed can be used to identify complex events (idle, startup, shutdown and in cycle) and the operational state of the machine. Example events are given in Table 3.1. Such an approach will also be used Section 6.5 < In semiconductor manufacturing, analysis of production-related data was shown to be capable of identifying significant energy saving potentials. Yu et al. [181] show how a neural network can be used to identify measures to reduce the energy demand of a semiconductor factory. The data that was used in the study by Yu et al. already represents aggregated time series data. Features like "mean process time" or "kWh per move" are used. They do not elaborate on how this data was acquired precisely, but the example illustrates how valuable it is to have aggregations like this available.

Gradišar et al. [70] present a framework that uses historical data to generate KPI-models using black-box methods. Those can then be used for an integrated production optimisation process. To ensure connectivity to existing software systems such as ERP, MES, and Supervisory control and data acquisition (SCADA), the framework includes a "data module", which is based on communication protocols and data representation standards used in the manufacturing domain. The authors present encouraging results from industrial use cases and confirm the approach, but also point out that existing software infrastructure often limits the applicability of the system. Their approach is in many regards very similar to the one described in Section 4.

The goal of fault detection in semiconductor- wafer production, is perused by Chien et al. [47]. Historical tool data in the form of time series sensor data together with quality data is used to define rules for fault classification. Among other necessary steps, data needs to be prepared by domain experts to identify the correct temporal windows and generate fault classes. Necessary information from other sources, such as production plans, as well as information regarding tool and product are not considered in the study, as this would require further data integration.

Recent work by Kang et al. [95, 96] combines data from the manufacturing process with data acquired from customers after delivery. As shown through a real-world use case, this approach improves the failure analysis for the products under investigation. Notably, the authors point out the importance of both data preprocessing and close cooperation with domain experts. Consequently, they advocate for the stronger incorporation of domain knowledge in the (automated) data mining workflow to improve its applicability and reduce the necessary efforts.

3.1.3 Limitations and Ways to Overcome Them

As a result of the last section, some common requirements and main issues for data mining in manufacturing domain can be identified:

- Time series data seems to be a crucial ingredient for many analyses that are carried out. This leads to problems which were to some extent already pointed out in Section 2.1
- The physical quantities that are represented by those time series, however, vary.
- The size of datasets often is a problem and limits the methods that can be applied. This is connected with the lack of data integration in the manufacturing domain mentioned by Bustillo et al. [33] and also Kusiak [112].
- Harding et al. [83] highlight the large amount of work that has to be put into the improvement of data quality and data preparation before any data mining can take place [83]. This is especially painful, as those that typically deal with the data might not have the necessary skills to do this preparation efficiently.
- Schuh et al. [154–156] as well as Reuter et al. [151, 150] mention the problem of data integrity and quality.
- Srinivas and Harding [167] point out the shortcomings of data mining applications for decision support in manufacturing specifically for shop floor control. Even though knowledge, according to them, today is the most valuable resource of any manufacturing firm, current knowledge-based systems fail to adequately store knowledge, which, as a consequence cannot be used in processes (including data mining).

To sum up, according to Kusiak [113], still today, there seems to be a lack of suitable software solutions that enable the use of data in the manufacturing domain. Companies still operate in isolation and research does not address practical problems enough. Among the innovation gaps, he identified is the need for improved data collection, use, and sharing. Especially data from the manufacturing domain is challenging due to the wide range of data frequencies.

In this respect, the potential of Semantic Web technologies in the manufacturing domain is surveyed by Ramos [148], who, however, focusses strongly on approaches that improve the interchangeability of product-related data between CAx software solutions. Several ontologies for the

manufacturing domain are presented and discussed. In general, the lack of methodical consistency and therefore reusability of the proposed ontologies is named as a limiting factor for their widespread use. Interestingly, on a side note, the need for a manufacturing process ontology is expressed as well. Recent studies [24] show the emerging trend of using Semantic Web technologies in industrial use cases, such as for simulation model generation [133], defect detection [110, 64], and various data integration scenarios [61]. The use of Semantic Web technologies for industrial simulation mostly focuses on employing an ontology as a common data model for integrating data from heterogeneous data sources for simulation purposes, such as Virtual Factory Framework [171].

To acquire industrial data as RDF graphs, which are normally required in Semantic Web technologies, two prominent approaches are (1) the traditional Extract, Transform, Load (ETL), which requires materialization of RDF graph over data sources, and (2) OBDA, which provides users with a virtual RDF graph over relational databases [37, 38] and other structured data sources, e.g., XML, CSV, and JavaScript Object Notation (JSON) [59]. Recently, novel approaches for integrating live data stream access, such as Ontology-Based Stream-Static Data Integration (OBSSDI) [103] and SPARQL stream extensions [72], have been proposed to extend the support for data acquisition. Different use of ontologies is proposed by Novák and Šindelář [132]. They are using ontologies to suggest and reason simulation model building blocks for simulation model development. These approaches, however, are developed based on the traditional ETL approach, which restricts the scalability of the system due to the limitation of the RDF graph storage capabilities [128]. Other interesting application areas for semantic technologies in the manufacturing domain, which are however not directly related to the scope of this thesis are standardisation [71] and cross-domain data integration.

Kharlamov et al. [100] discuss potentials of OBDA in large companies illustrated by the application of this technology at Siemens Energy, where several service centres for power plants are in operation. Data is used to perform several monitoring and diagnosis tasks. The bottleneck in any data analysis process is the gathering of data, consuming approximately 80% of the overall time needed to accomplish the task. This is, according to the authors, "*due to the mismatch between the language and structures that the engineers use to describe the data and the way the data is described and structured in databases*". OBDA is expected to be a suitable way to overcome this. In Calvanese et al. [40], the current limitations of OBDA are described. At the moment, OBDA systems are not able to reflect the temporal nature of data sufficiently. This has a significant effect on the modelling capabilities also in the production domain. For example, the identification of abnormal situations occurring in connection with devices based on monitoring data is named as a likely use-case for temporal ontologies in the field of mechanical or electronic devices. According to the authors, further work in this field using real-world data will be done in the future.

Within the manufacturing domain, OBDA has successfully been used by, for example, Petersen et al. [142]. In their work, they report on a case study, in which they realised an information model for a global manufacturing company based on RDF vocabularies. Apart from conventional triple stores, they provide a domain-specific system architecture that integrates data concerned with workpieces

(BOM), work orders and manufacturing processes (MES), sensory data (which are stored in RDBMS) and geological data (which is stored in RDF-format).

They show some example applications that the company was interested in and evaluate the performance of the system based on qualitative interviews with stakeholders from the company. The first example of an application is concerned with tool management and integrates data from MES with geological data to determine tool availability. In their second example, they showcase how sensor data in combination with MES and BOM data can be used to answer queries regarding energy demand of production processes.

Even though their results are encouraging, it also illustrates the current shortcomings of available tools with respect to the requirements within the manufacturing domain. While data from MES, BOM can be accessed and mapped to ontological concepts relatively easy, the access to (temporal) sensory data still seems to be rather complicated. This is because available languages mostly lack the expressivity to address temporal relations between different concepts. However, before approaches to this are more closely reviewed, current developments in the area of OBDA will be shown in the following Section.

3.2 Ontology-based Data Access

Figure 3.4 shows an overview on the field based on publications in the field of OBDA[1].

[1]Query Details: Date: 23.11.2018; Query: TITLE-ABS-KEY (("OBDA" OR "Ontology-based Data Access")) ; Source:https://www.scopus.com

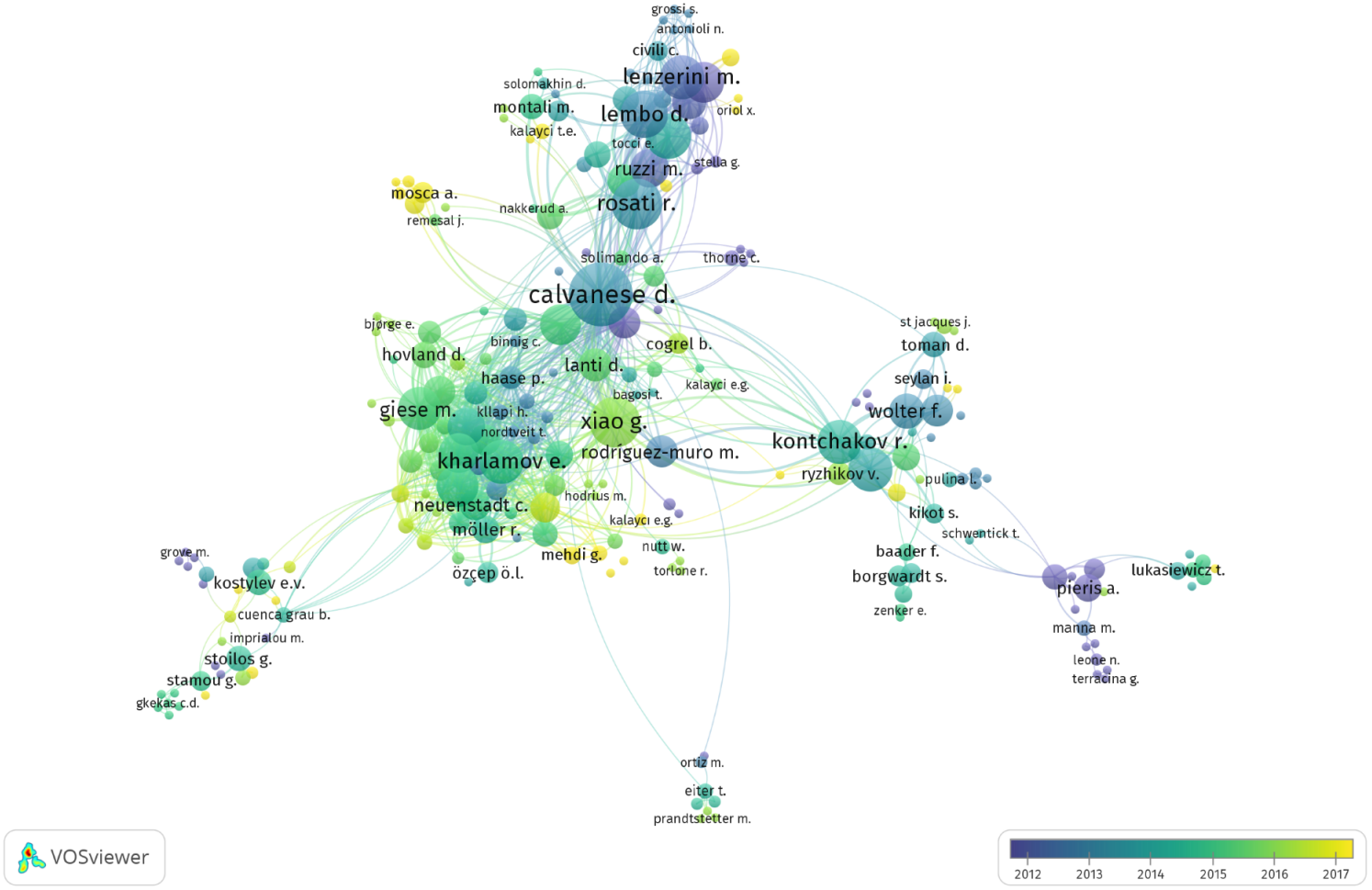


Figure 3.4 Publication overview for OBDA based on query from Scopus. In total 418 publications were found. Clusters are created based on co-authorships. Bubble size is based on number of total documents within the query result.

Poggi et al. [144] first introduced OBDA. Using the logical foundations provided in the form of $DL-Lite$ [37], they present a way to create mappings between ontologies expressed in this language and data that is stored in relational databases. Also, they describe a query answering method that uses the ontology and the mappings to the data layer. First, a query is expanded according to the knowledge expressed by the ontology. The result then is unfolded, i.e., translated, using the mapping into a query to the actual data storage. With QuOnto, Owlgres and REQUIEM, first implementations of OBDA were developed as well [3].

The OBDA- approach was also taken up by other researchers, such as Calì et al. [34, 35]. In their work, they propose a framework that uses tuple-generating dependencies to represent ontological constraints in order to overcome complexity limitations of $DL-Lite$ while still retaining low complexity of reasoning. Until recently, however, no system was developed that did put this alternative approach into practice, which explains why this line of research was not yet continued.

Based on these developments, several researchers at that time affiliated with the University of Rome worked on this topic. There, among other things, the Mastro system was developed [38]. This system was an implementation of the principles mentioned above. Developed in Java, it can be connected with a variety of different data sources through a Java Database Connectivity (JDBC) connection. Furthermore, it provided a plugin for Protégé, which is a popular ontology editor. A further step towards commercialisation of Mastro was taken by Civili et al. [54] in the form of Mastro Studio, which offers improved user interfaces.

A shortcoming of the initially proposed rewriting approach that was implemented in previous tools was addressed by Kontchakov et al. [107]. Specifically, the problem was that queries after rewriting became too big to be handled by an RDBMS [38]. As an alternative, they chose a combined approach. Instead of adding knowledge to the query and then unfolding it (conventional approach), two other steps are executed. First, the data that should be queried is extended taking into account the TBox. Any query then is rewritten to a query over this expanded data source. The first step, however, led to the discovery of a significant limitation- the solution required the ability to change the data source, which often is not available.

An essential step towards the maturity of OBDA was the Optique project [80], which was funded by the European Union and brought together the majority of researchers in the field. The goal of this project was to provide an end-to-end solution for OBDA to Big Data integration. Two industrial use-cases motivated the development of this platform. One of which is the use case described by Kharlamov et al. [100], also briefly mentioned in the introduction (Chapter 1) of this thesis.

Classical OBDA does not support access to or reasoning over temporal data. As was observed in the course of the Optique project, this poses a significant limitation to many relevant application scenarios. Therefore, recently, the potential for temporal description logics in the field of OBDA has seen much attention from the scientific community. In their statement of interest, Horrocks et al. [87] lay out the requirements that result from the use cases within the Optique project. Especially in the case of Siemens, the authors argue that reasoning over both historical and streaming time series data requires the ability to consider temporal concepts. Specifically, they formulate requirements

for temporal operators to express things such as "always in the past". Furthermore, the capability to use stream-oriented operators such as sliding windows is expressed. Finally, aggregation functions (time series analysis operators, as they call it) such as "mean" or "standard deviation" are mentioned. They point out that existing solutions do not live up to their requirements either due to their limited functionality to reason over temporal concepts or due to the missing integration of functionality in one tool. As a consequence, they describe how they plan to approach this problem, which is through an extension of RDF with validity times. This results in a large number of promising approaches using different techniques, which will be described in the next section.

3.3 Temporal Ontology-based Data Access

OBDA, as it was described until now, does not provide ways to consider temporal aspects of data natively. Based on concrete OBDA use cases from industrial applications, however, the ability to consider temporal aspects of data was identified to be crucial for practical applications [40]. This judgment, however, is not limited to researchers from the Optique project. Also, other authors emphasise this requirement [60].

An illustrative example that motivates the development of temporal OBDA is given by Calvanese et al. [40]. Consider a set of weather stations measuring climate conditions (temperatures, humidity, wind speed) at different places and thereby generating time series data. This data can be used for a variety of analysis. For example, all detections of hurricanes might be of interest. A hurricane is detected when measurements read wind speeds that exceed 118 km/h for one hour. In order to capture this definition, it would be beneficial to be able to use temporal constructors for concepts. In order to keep the resulting solutions practically usable, the temporal constructors need to be limited to those that are necessary for a given application area. This makes it necessary to integrate experts from the problem domain under consideration much more closely into the ontology development and DL selection process.

As Calvanese et al. describe, there are more than one ways to do this [40]:

1. Temporal Extension of the Ontology Language: [10, 108, 111]
2. Temporal Mapping Languages: [93]
3. Temporal Query Language: [78, 15, 26, 105, 136, 137, 9]

Figure 3.5 shows a visualization of a query[2] for publications in the field of temporal OBDA. In this case, however, only those publications that were concerned with temporal aspects were taken into account.

As can be seen, there are a few distinct clusters that can be identified. The first cluster at the bottom left corner of Figure 3.5 represents papers regarding the Siemens use case from the Optique

[2]Query Details: Date: 23.11.2018; Query: TITLE-ABS-KEY (("OBDA" OR "Ontology-based Data Access") AND "Temporal*") Source: Scopus; Results:41; Source:https://www.scopus.com

project, based on the position paper by Horrocks et al. [87] presented in the previous section, several different application oriented developments originated [123, 69, 101, 163, 165, 104].

Another cluster, which shows a second path of origin that is connected with another field of research, can be seen around Calvanese et al. [39]. There exists a connection between the field of process mining and temporal OBDA [39, 84, 40, 42]. This cluster, however, is not directly related to the field of OBDA and will therefore not be described in further detail.

More recent developments can be separated into two clusters. The first one is in between the two clusters mentioned above. (Mainly) Researchers from the institutions that participated in the Optique project seem to carry on working on the topic by both developing concrete applications and extending the logical foundations necessary [9, 108, 40, 28, 29].
Another group of researchers around Baader and Borgwardt is also working on the field. They do, however, mainly focus on extending query languages with temporal concepts [15, 26, 16, 17, 27]. Finally, there are a few "outliers" that are working on similar topics but have no record of co-authoring papers in the field with other researchers that are part of the groups mentioned above [105, 98, 78].

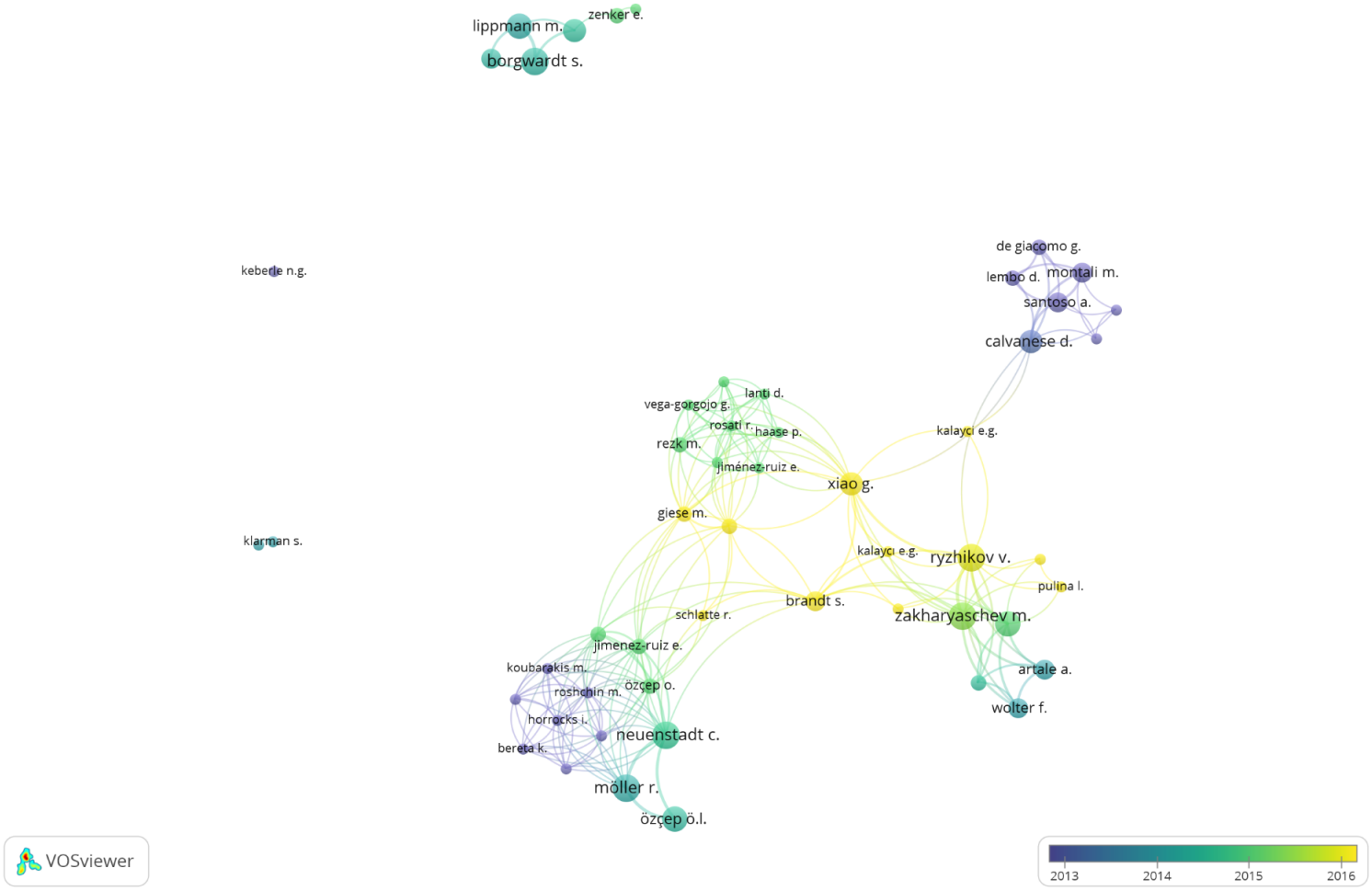


Figure 3.5 Publication overview for Temporal OBDA based on query from Scopus. In total 56 publications were found. Clusters are created based on co-authorships. Bubble size is based on number of total documents within the query result.

3.3.1 Cluster 1: The Optique Project

Already, the Optique project was mentioned. As a consequence of the size of the project, a considerable number of publications in the field originated from this project. To cope with the requirements from one of their use cases, namely stream reasoning requirements were tackled through the development of a new query language framework, STARQL [136, 137, 131]. STARQL is syntactically similar to SPARQL but allows to include temporal arguments in queries. The general approach, which allows for various query languages and DLs to be applied, is illustrated in the specific case of $DL-Lite$ and unions of conjunctive queries, which are especially important in the OBDA context.

Möller et al. [123] show the performance of STARQL through experiments with an implementation of the framework. They create mappings between a subset of the data from the Siemens use case (5-23.000 MB), which is stored in the PostgreSQL DBMS and their (temporal) series of ABox assertions. Then two queries witch different complexity were used to evaluate the performance of the system. Especially for "large" datasets, the query times are in the order of several minutes, which raises the question of the feasibility of this approach.

Based on these results, an alternative application scenario is described by Kharlamov et al. [102]. The authors propose to use results such as the ones that they developed, as an Ontology- based Data Integration (OBDI)- approach to solving the problem of accessing stream data that is stored in distributed databases. A common global ontology is defined and mapped to local databases. Queries are then executed with respect to the ontology. The fact that STARQL enables users to combine static with temporal data corresponds with requirements from real-world application scenarios.

Another extension of STARQL is described by Kharlamov et al. [101]. To improve the usefulness of OBDA for analytical tasks, the authors present an extension that makes it possible to include aggregation function into queries. To do so, they facilitate the ontology language $DL-Lite_{\mathcal{A}}^{agg}$, which is an extension of $DL-Lite_{\mathcal{A}}$ and allows to define concepts that are based on aggregation functions (min, max, count, sum or avg) of attribute values. Furthermore, comparisons ($\leq, \geq, =, <, >, \neq$) can be used.

Another, from the user perspective, significant development in this branch of research was the development of an interface for visual query formulation, which further improves the usability of the developed system Soylu et al. [164, 163, 165]. They propose an extension of existing ontology-based visual query formulation tools with the ability to also formulate temporal queries coining the term ontology-based visual stream-temporal querying. The developed system is also evaluated through user experiments. Three users were given queries that typically occurred in their respective fields and asked how comfortable they were formulating them. The feedback as well as the quality of the formulated queries are very encouraging and indicate that such a system can significantly reduce the time demand for data access tasks.

3.3.2 Cluster 2: TU Dresden

Also, researchers from outside of the Optique consortium are concerned with the topic of temporal OBDA. One of the first articles on this topic was published by Baader et al. [15]. Sensors give (incomplete) information regarding an arbitrary system. The signals are stored in a fact base, which often is implemented in the form of a relational database. These facts can then be processed by, for example, a situation awareness application. As a motivating scenario, they give an example from the medical domain. Based on heart rate measurements, the state of a patient can be determined. In order for such a system to be feasible, however, temporal aspects need to be taken into consideration. To realise this, they propose a combination of a static T-Box with a series of A-Boxes that reflect the state of a system for each discrete point in time. For this, they build on LTL.

LTL is a logic introduced by Artale et al. [8] and provides operators to express "sometime in the future" ($\Diamond_F$) and "sometime in the past" ($\Diamond_P$),"always in the future" ($\Box_F$) and "always in the past" ($\Box_P$) as well as "next" ($\circ_F$) and "previous" ($\circ_P$). In this approach, developments over time are seen as a sequence of ABoxes $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n$ with $0, 1, \ldots, n-1$ representing previous timepoints. This is a slightly different approach than the one chosen by Özçep et al. [136], who use timestamped ABox assertions. Based on this notion, they develop complexity results for the case of an $\mathcal{ALC}$ TBox. Remarkably, those do not differ from those of comparable atemporal cases. In their outlook, they express interest in doing similar investigations for less expressive description logics such as $DL-Lite$.

Using LTL, conjunctive queries can be created that also take into consideration temporal aspects. In their approach, Baader et al. [14] allow the temporal component to influence queries via rigid names that cannot change over time. Building on their previous work, the authors use LTL as a temporal query language that provides the ability to use constructs such as "since" or "sometime in the past" to query data in an OBDA context.

More recently, Baader et al. [16] developed a temporal extension of a more expressive logic ($\mathcal{SHQ}$ [16] and $\mathcal{SHOIQ}$) as a query language and investigated complexity results. Notably, the upper and lower bounds that were found mostly coincide with those of atemporal equivalents of respective logics.

3.3.3 Cluster 3: Post- Optique Developments

Borgwardt et al. [27] follow up on the plan to combine temporal logics with less expressive DLs such as $DL-Lite$. This is motivated by a scenario where past monitoring data shall be used to recommend actions in the future, which initially seems to make not only operators for "past" but also "future" necessary. First, the authors show how to rewrite temporal queries towards a $DL-Lite$ KB to an equivalent query towards a temporal database. Due to the temporal nature of this task, some complications arise. Space needed to store data from all previous timepoints makes it infeasible due to the high frequency at which new data is generated. This makes it necessary to identify only those parts of the data that are needed for query answering and neglect the rest. The authors investigate

different ways to approach this and finally present an algorithm that does not only solve the issue of possible query rewriting but also can be used to improve the expressivity of the used KB.

Another approach towards improving OBDA temporal data through temporising query languages is described by Artale et al. [9]. They investigate how the first-order rewritability of different variations of $DL-Lite$ behaves in combination with different LTL operators.

Brandt et al. make an effort to take previous developments to the next level by proposing a framework for temporalised OBDA [29]. First, they expand "conventional" OBDA systems as described above (Section 3.2) by adding a temporal vocabulary to both the ontology $\mathcal{O}$ ($\mathcal{O}_t$) and the mapping $\mathcal{M}$ ($\mathcal{M}_t$). Note, that this goes well beyond what was done in previous work which only included temporal aspects to the query language. The authors point out that this approach is again shifting the load of handling temporal data to the user and find that current, $DL-Lite$-based OBDA systems are not expressive enough to capture the concepts that are interesting from the practical point of view. Consequently, they propose $datalog_{nr}MTL$ as a language to capture domain kknowledge [108, 28]. Datalog is a declarative logic programming language, which is used to query deductive databases, which were developed to expand relational databases with logic. MTL provides constructs that make it possible to capture periods and explicitly state their length. In regards to mappings, the authors propose to extend mappings as they are used in conventional OBDA-systems with the ability to also state validity intervals for mappings. Finally, the authors propose $\mathcal{T}$-SPARQL [170] as a means to query data. This query language extends well-known SPARQL with the validity intervals that can be added to the query in the form of a prefix.

As the most recent continuation of the work conducted in this line of research, Kalaycı and Calvanese proposed a different approach to temporalising OBDA based on Ontop, which involves temporal concepts into the mapping between ontologies and the data source [93]. This system, Ontop Temporal, which currently represents de facto the only available tool for temporal OBDA, will be used for the implementation of a manufacturing prototype in Chapter 6.

3.3.4 Outliers

Apart from the larger clusters, some researchers without an obvious connection to the rest of the community did some interesting work on temporal OBDA as well. In an attempt to unite several different research directions and thereby define an architecture that is suited for most applications, Gutiérrez-Basulto and Klarman [78] introduce a basic framework for representing temporal data in DLs by using a series of time-stamped ABox assertions. The timestamps indicate the validity of each ABox. Furthermore, they propose a mechanism for defining temporal query languages. Other authors further applied this approach.

A similar approach to the temporal aspect is taken by Klarman and Meyer [105]. Following the then-new standardization of the SQL query language SQL:2011, they propose the interval-based Temporal Query Language (TQL) to bridge between this new standard and OWL 2 QL, which is the most common profile for OBDA. Following the SQL standard, validity periods can be assigned for

facts that are stored in database tables. These periods than can be used through TQL. A concrete example for such a query would be "Find all persons X and times Y, such that X worked in a department based in Barcelona during Y and in a department based in Madrid sometime earlier."

$\mathcal{EL}$ description logic is, for example, used heavily in the medical domain. Gutiérrez-Basulto et al. [79], investigate a temporal extension of this description logic with respect to reasoning complexity and query rewritability. Even though unrestricted versions turn out to be undecidable, reasonable results can be obtained for smaller fragments which, for example, only allow for "next" or "previous" operators [13]. Lécué and Pan [116] use the concept of *ontology stream reasoning* to reflect changing knowledge over time. Motivated by the goal to improve predictions made by machine learning approaches, they investigate possibilities to predict knowledge in an ontology stream in order to be able to capture temporal phenomena. In this approach, even though the KB is evolving, i.e., subsumptions and assertions are changing over time, the axioms themselves do not contain temporal constructs.

Chapter 4

Application Scenario: Energy Center

As stated before, data analysis, especially of time series data, is a method that is becoming increasingly popular in the manufacturing domain. In this chapter, an example use case from the research project BaMa [1] is described. For the given use case, a PoC, using OBDA is presented as a candidate for such a measure and evaluated through user feedback.

The goal of the BaMa project was to develop and implement a simulation-based method for monitoring, predicting and optimising energy and resource demands of manufacturing companies under consideration of the economic success factors time, costs and quality [118, 86].

Within the BaMa project, 18 partners collaborated to generate solutions for concrete use cases as well as the necessary scientific methods and technological framework to implement them. One of those partners was Infineon Austria AG (INF), which provided two different use cases. One of these use cases is concerned with the optimal operation of the machines that are used to provide thermal energy in order to keep clean room conditions within one of their production facilities [129].

The semiconductor industry heavily relies on production in cleanroom conditions. A cleanroom is defined in the ISO standard 14644-1 as:

"Room in which the concentration of airborne particles is controlled, and which is constructed and used in a manner to minimise the introduction, generation and retention of particles inside the room and in which other relevant parameters, e.g. temperature, humidity and pressure are controlled as necessary." ([89])

In general, to maintain those conditions, large amounts of energy in the form of heat and cold need to be provided to an air-conditioning system. This is the purpose of the energy center, which is typically located on the premise of the production site.

The system under investigation, which is an instance of such an energy center, is depicted in Figure 4.1. Twelve chillers with nominal electrical power input rates between 360 kW and 500 kW were investigated. Energy demand is determined mainly by the load level of the production plant. There, an air condition system is used to keep the room climate within specified limits — the required energy demand changes due to fluctuations in the waste heat from production machines and changing

[1] http://bama.ift.tuwien.ac.at

ambient conditions. Thermal energy, both heat and cold, is transported in the form of water flowing in separated energy grids.

Industrial chillers are used to convert some form of energy to cold. In the case of compression chillers, electrical energy is used. In order for this to be possible, those machines need to have access to some heat sink, which, in this case, is mainly constituted of cooling towers. Some of the chillers, however, are equipped with heat- recovery systems which enable them to not only provide cold but also heat to the production plants. Any heat that can not be provided by these systems is taken from an external source, which leads to additional costs.

Before the BaMa project, the control strategy of the plant worked as follows (for simplicity, only cold is considered in this explanation): The water temperature in the girds (approximately 6 °C) is a measure for the amount of energy that is available at any given moment. For every load level, there is a corresponding number of necessary chillers that are needed to cover it. If changes in the load level profile occur, the temperature in the energy grid increases or decreases and when a threshold is passed, the number of chillers in operation is changed by switching them on or off accordingly. The respective chillers are chosen by their accumulated operation hours. The goal is to evenly distribute the operating hours on all machines so that simultaneous maintenance is possible. Furthermore, each chiller has an operating range which can be used to react to smaller fluctuations.

The goal of this application scenario was to improve the operating strategies for the energy center considering total cost, energy demand and CO_2- emissions while still providing the required heating and cooling power to production plants and office buildings.

Simulation-based optimisation is widely accepted to be an appropriate solution for the improvement of the performance of large industrial chiller plants. Chua et al. [53], for example, present a system for smart chiller sequencing. Beghi et al. [21] propose a multi-phase genetic algorithm to improve the performance of a multi-chiller system under consideration fo the required electrical power input. Another approach was proposed by Askarzadeh et al. They used particle swarm optimisation to reduce the electrical power consumption of a multi-chiller system. Through particle swarm optimisation in combination with neural nets, which were used to provide predictions regarding energy demand, Chen et al. achieved a 17% reduction of required electrical energy demand for a multi-chiller system.

In this case, optimisation potential was expected to exist due to the wide variety of chillers available. Depending on age and machine type, the model of environmental factors, every chiller that can be used has its unique operation characteristic, which was initially unknown. Therefore, in order to find optimal operation strategies, following the presented BaMa- approach, a simulation model of the energy central was created. This was done using available monitoring data from the deployed chillers and will be described in further detail in the next section (Section 4.1). This simulation model then was coupled with an optimisation module. The results and corresponding optimization potentials are described in section 4.3.

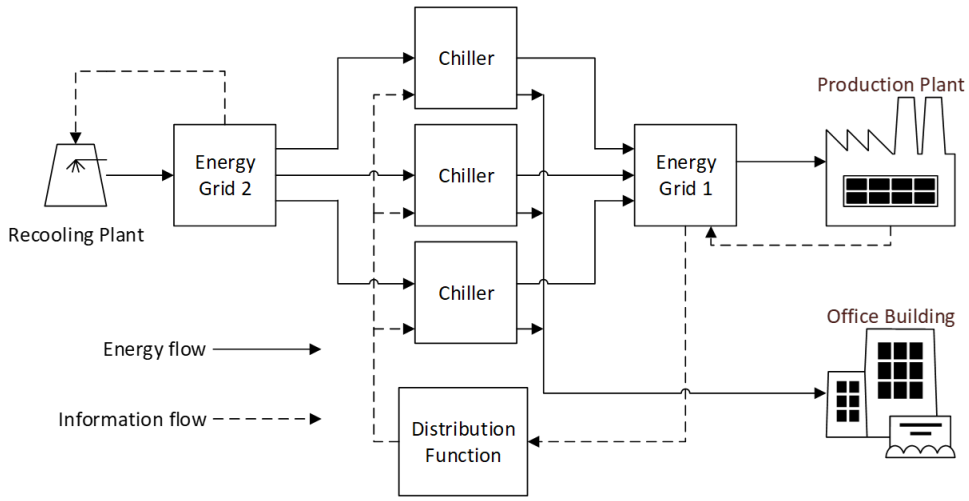


Figure 4.1 Structure of the energy center under consideration [65].

4.1 Modelling Approach

In order to create a simulation model of the whole facility, relevant subsystems are identified and combined to result in a model of the reality. The multi-physics simulation tool AMESIM[2]was used for the implementation of these models. Subsystems represent chillers, energy grids and cooling towers. Subsystem models are connected via interfaces. These connections are depicted as lines in Figure 4.1.

Data representing energy (full lines) and control information (dotted lines) can be exchanged. The primary input to the simulation, which is determined by the respective scenario that motivates the optimisation run, is the total cooling demand from the production facility ($\dot{Q}_{C,d}$). Based on this information, the cold grid models seek to provide the requested energy demand by sending the requested energy flow ($\dot{Q}_K$) to the production facility. This reduces the amount of stored energy (if any energy is stored) within them. Based on the internal controller of the grid and a distribution function, which determines the order in which new chillers are switched on and off, the grid then requests energy from chillers. The distribution function determines how a given energy demand from the grid is distributed to the available chillers.

Both, heat recovery and non-heat recovery chillers need to be provided with cooling water themselves. This cooling water is provided for by cooling towers, which use vaporisation to emit heat to the environment. The efficiency and capacity of cooling towers are mainly determined by environmental conditions (humidity and temperature).

As can be seen from this description of the overall system, many different subsystems exist. Apart from the industrial chillers, which by far played the most crucial role in the overall performance, those where the cooling towers and energy grids.

Gray-box modelling was chosen as modelling paradigm in this use case. It is a combination of white-box (analytical formulas) and black-box (mathematical models trained on empirical data)

[2]https://www.plm.automation.siemens.com/

Figure 4.2 Industrial Chiller, representative for those in the use case.

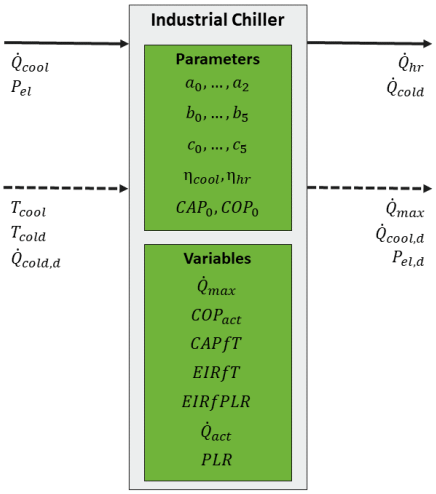


Figure 4.3 Model overview. Full lines represent energy, dotted lines information interfaces.

approaches and therefore are a to describe system behaviour. They are a trade-off between the implementation simplicity of white and the general applicability of black box approaches.

Parameters are used to describe the behaviour of instances of abstract models. Inputs representing energy and information flows are processed internally, via variables. Eventually, this generates outputs that can be sent on to the subsequent system models. In the following subsections, the respective simulation models for each of the subsystems will be described.

4.1.1 Industrial Chillers

The chillers considered in this use case are compression chillers such as the one depicted in Figure 4.2.

A grey-box approach was chosen to model the performance of industrial chillers. Specifically, the chosen model (depicted in Figure 4.3) utilises a set of polynomial functions to describe the behaviour of the chillers [134]. This particular model considers the influence of the cold water temperature (T_{cold}), the cooling water temperature (T_{cool}) and the part-load ratio (PLR) of the chiller on the efficiency and the capacity of the chiller. Efficiency (COP_{act}) can be interpreted as the ratio between required electrical power and cooling power. Capacity ($\dot{Q}_{max}$) is a measure for the amount of maximum cooling power that can be provided by the chiller. These measures depended on the temperature of the cold water (T_{cold}) at the entry of the chiller before it is cooled down and distributed to the air conditioning systems. Cooling water temperature (T_{cool}) is the temperature of the water that is reaching the chiller from the cooling towers. Both measures, efficiency and capacity, can be used to calculate the electrical power input demand for each chiller ($P_{el,d}$) (4.1), which is a core element of the optimisation of the overall system and therefore also the primary output of the model.

$$P_{el,d} = \dot{Q}_{max} \cdot COP_{act} \tag{4.1}$$

To determine $\dot{Q}_{max}$ (4.2a), the nominal cooling capacity (CAP_0), which is provided by vendors for each chiller, is multiplied with a bi-quadratic polynomial ($CAPFT$) (4.2b), which describes the influence of the temperature of the cold water (T_{cold}) and the cooling water (T_{cool}) on the capacity of the chiller.

$$\dot{Q}_{max} = CAP_0 \cdot CAPfT \tag{4.2a}$$

$$CAPfT = c_0 + c_1 \cdot T_{cold} + c_2 \cdot T_{cold}^2 + c_3 \cdot T_{cool} + c_4 \cdot T_{cool}^2 + c_5 \cdot T_{cool} \cdot T_{cold} \tag{4.2b}$$

COP_{act} (4.3a) is determined by the temperatures (4.3b) as well as the part-load ratio (PLR) (4.3c). The part-load ratio is the ratio between actually provided cooling power ($\dot{Q}_{act}$) and the theoretical capacity of the chiller ($\dot{Q}_{max}$) under the current conditions (4.3d). The maximum power that can be provided by a chiller ($\dot{Q}_{act}$) is, by definition, limited by its capacity ($\dot{Q}_{max}$). As long as the demanded power is below that threshold, however, the provided power is equal to the demanded power ($\dot{Q}_{cool,d}$) (4.3e).

$$COP_{act} = COP_0 \cdot EIRfT \cdot EIRfPLR \tag{4.3a}$$

$$EIRfT = b_0 + b_1 \cdot T_{cold} + b_2 \cdot T_{cold}^2 + b_3 \cdot T_{cool} + b_4 \cdot T_{cold}^2 + b_5 \cdot T_{cool} \cdot T_{cold} \tag{4.3b}$$

$$EIRfPLR = a_0 + a_1 \cdot PLR + a_2 \cdot PLR^2 \tag{4.3c}$$

$$PLR = \frac{|\dot{Q}_{act}|}{\dot{Q}_{max}} \tag{4.3d}$$

$$\dot{Q}_{act} = min(\dot{Q}_{cold,d}, \dot{Q}_{max}) \tag{4.3e}$$

Until now, the model provided by [134] was sufficient. Another output was needed, however, to be able to connect the chillers models with those representing the cooling towers. The cooling power demand ($\dot{Q}_{cool,d}$), is calculated using the following simple model (4.4).

$$\dot{Q}_{cool,d} = \eta_{cool} \cdot (\dot{Q}_{act} + P_{el,d}) \tag{4.4}$$

Furthermore, the model does not cover heat recovery ($\dot{Q}_{hr}$). Therefore, using a similar approach, also this had to be added (4.5).

$$\dot{Q}_{hr} = \eta_{hr} \cdot \dot{Q}_{act} \tag{4.5}$$

In order for the model to be representative for a specific chiller, the free parameters ($a_0, \ldots, a_2$, $b_0, \ldots, b_5$, $c_0, \ldots, c_5$, η_{cool}, η_{wrg}) needed to be estimated, which in this case was done based on historical monitoring data. This process will be described in further detail in section 4.2.

4.1.2 Energy Grids

As can be seen, chillers are not directly connected with the air condition system or the cooling towers. Between those systems are energy grids, which are used to distribute the energy according to some control strategy. Furthermore, energy grids are the only systems that have temporal behaviour, acting as energy storages, which makes it necessary to model them using differential equations. An overview of the chosen model is depicted in Figure 4.4. The amount of energy that a given grid can store is defined by its capacity (C), which is a function of the total mass of water (m) in the grid (which is constant) and the specific heat capacity of water (c_p) (4.6).

$$C = m \cdot c_p \tag{4.6}$$

The capacity is used to describe the amount of energy stored as sensible heat at a given temperature level within the grid (4.7a). The grid temperature (T) is also used to calculate values for T_{cool} and T_{cold} used above. Using the energy balance, the development of the grid storage state and the net heat flow in respect to the grid can be established (4.7b). The net heat flow ($\dot{Q}_{net}$) is the sum of all heat flows to ($\dot{Q}_i$) and from the grid ($\dot{Q}_j$) (4.7c).

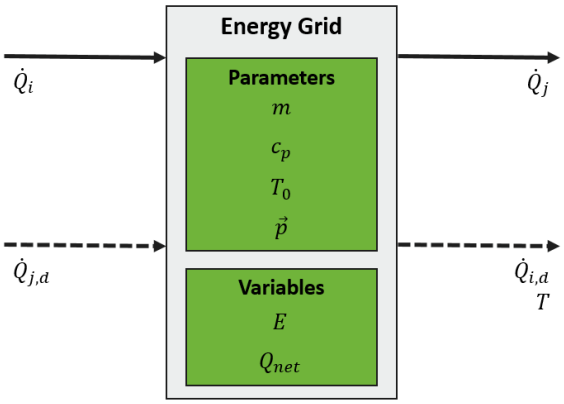


Figure 4.4 Energy grid model overview. Full lines represent energy, dotted lines information interfaces.

$$T = \frac{E}{C} + T_0 \tag{4.7a}$$

$$\frac{dE}{dt} = \dot{Q}_{net} \tag{4.7b}$$

$$\dot{Q}_{net} = \sum_i \dot{Q}_i - \sum_j \dot{Q}_j \tag{4.7c}$$

In reality, as well as in the simulation, the grid temperature is subject to a controller, which can in principle work according to arbitrary strategies. In any case, the task of the controller is to keep the temperature within the grid constant. In order to achieve this in the presence of changing conditions (such as varying demand, but also heat loss), this controller changes the heat that the grid itself demands from its suppliers. Also, this has to follow a particular strategy, which determines the order in which additional power is requested from individual suppliers by the grid.

Take, for example, the energy grid connecting the production facilities air condition with the industrial chillers. If, at a given moment due to an increase in production output also the required cooling power increases, first the temperature within the grid will start to increase. Depending on the controller, this will increase the power that is demanded from the currently active chillers. If those chillers have enough capacity left (i.e. a low enough *PLR*), their utilisation will increase. If, however, all active chillers are at some point running at their current maximum capacity, additional chillers need to be activated. A distribution function determines the selection of those chillers. The same process applies for sinking power demand and the consequent switching off of chillers.

At the core of this distribution function is a priority vector ($\vec{p}$). This priority vector is what was subject to manipulation throughout the optimisation process, which is described further in Section 4.3.

4.1.3 Cooling Towers

In the case of the cooling towers, there exists a strong relationship between the capacity of the cooling towers and the prevalent weather conditions. This relationship, however, could not be modelled due to a lack of available data. Therefore, a fixed maximum capacity was chosen. Similar to what was described in (4.3e), the cooling towers can only provide cooling power until their capacity threshold is reached. In the case that this is below what is requested from the energy grid, the cooling towers cannot follow the request and consequently the grid temperature rises, which leads to a deterioration in chiller performance.

4.2 Parameter Estimation

The presented simulation models depend on parameters. Some of those parameters could be retrieved from system specifications (i.e. in the case of the energy grids), for others, especially the chillers, they had to be estimated from historical time series. In the following section, the process of estimating the parameters for the chiller model will, therefore, be described.

4.2.1 Data Source

For all considered chillers, data is stored in RDBMS at INF among monitoring data from other systems. This data had to be transferred to the scientists in charge of building the simulation models. Direct access to the respective databases, however, could not be granted due to security reasons. Therefore, verbal requests from scientists were formulated and sent to the contact person at INF. Example queries where: "We need all cold water temperature measurements of chillers within building XY in the period from June 2016 to December 2016". This request had to be further processed by the engineers. For example, the exact names of chillers located in a specific building had to be retrieved from printed-out schemas. The processed queries were then handed over to an IT expert, who translated the request to a SQL-query. The result of this query was then forwarded to the scientists in the form of CSV-dumps.

Table 4.1 Model errors per chiller after parametrisation

	data set size		RMSE [%]	
Chiller	**without filter**	**with filter**	P_{el}	Q_{cool}
Chiller 1	919,865	919,757	13.6	65.1
Chiller 2	1,124,472	1,124,415	17.2	4.5
Chiller 3	1,125,003	1,124,808	4.7	10.0
Chiller 4	1,125,671	1,125,611	26.9	21.2
Chiller 5	915,289	915,143	38.5	48.6
Chiller 6	1,134,663	1,134,474	178.3	722.2
Chiller 7	1,127,933	844,476	126.1	136.2
Chiller 8	678,718	583,815	28.8	69.1
Chiller 9	679,007	587,077	10.3	3.9
Chiller 10	679,250	591,193	4.4	8.4
Chiller 11	679,451	556,556	11.8	29.7
Chiller 12	103,660	74,159	179.5	432.2

Often, query results again had to be processed in order to become useful for the parametrisation process. Often, irrelevant measurements were included, and relevant measurements were missing. Furthermore, the meaning of the column names concerning the initial request had to be investigated more than once.

An overview of chiller related data used for parametrisation is given in Table 4.1. For a total number of twelve chillers, sufficient historic data (records of 30s averages of electrical power input and cooling water/ cold water power equivalent) was available to find model parameters. The respective data sets were preprocessed based on heuristic criteria formulated by domain experts (see Section 4.2.2). The datasets per chiller had between 70,000 and 1,000,000 samples and were therefore sufficient for the estimation of the required model parameters. For four of these chillers, also temperature measurements were available. Two chillers had to be excluded from the optimisation process due to their exceptionally high model errors.

4.2.2 Parametrization Approach

Based on the available data, the following process was developed to identify the chiller model coefficients by using collected monitoring data. The implementation in MATLAB can be found in Appendix A.

1. Load data: Script loads the data from CSV files and creates MATLAB tables for further processing.

2. Filter Outliers: This step improves the overall model quality by omitting outliers and incomplete data. Data instances were removed if either one of the measured attributes was missing or measurements lay beyond threefold the standard deviation or any data containing temperature

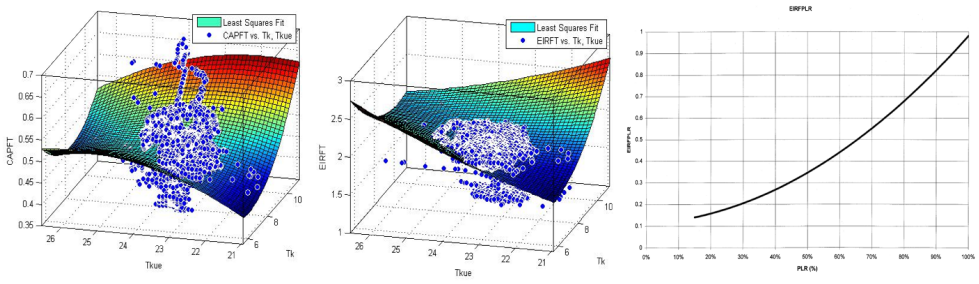


Figure 4.5 Depiction of the model fitting process. Data points are interpolated using (bi-) quadratic functions. The mismatch between data and the function is apparent and illustrates the importance of data preprocessing.

signals outside of defined thresholds ($4\,^{\circ}\text{C} \leq T_{cold} \leq 15\,^{\circ}\text{C}$ and $15\,^{\circ}\text{C} \leq T_{cool} \leq 30\,^{\circ}\text{C}$). The effect of this filtering step can also be found in Table 4.1.

3. Split dataset: The resulting, cleaned dataset was split into two parts. 90% of the data were used for training, 10% were used for validation of the model parameters.

4. Parameter estimation: Based on the work of Monfet and Zmeureanu, model parameters were estimated by using least squares interpolation on the available monitoring data [124–126]. Examples of the resulting interpolation functions ($EIRfT, EIRfPLR, CAPfT$) are depicted in Figure 4.5

5. Parameter validation: Using cross-validation, the average model error for each parameter set was calculated using the root- mean square error (Equation 4.8)

$$RMSE = \frac{\sqrt{\sum_{i=1}^{n}(y_{i,predicted} - y_i)^2}}{\sum_{i=1}^{n} y_i} \cdot 100 \tag{4.8}$$

Based on the analysis of the model errors, nine chillers were selected for optimisation (Table 3).

4.3 Optimization

Until now, the structure and the parametrisation of the simulation models representing the relevant subsystems for the use case were described. In the following section, the optimisation process will be described which facilitated those simulation models to generate optimal operation strategies will be described.

4.3.1 Optimization Approach

The general optimisation approach concept is illustrated in Figure 4.6. As can be seen, simulation is at the centre of the optimisation process. First, an arbitrary scenario is defined and modelled in

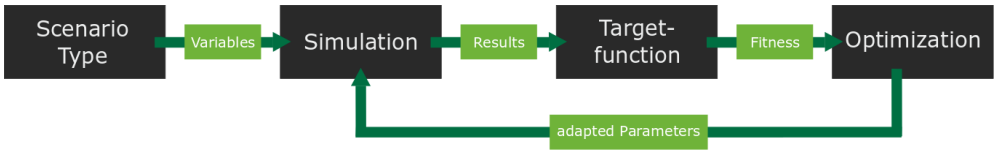


Figure 4.6 Conceptual view on the interaction between simulation and optimization in the BaMa-Project.

the form of variables with associated constraints so that it can be processed by the optimisation tool. Variables, in this case, are both inputs and outputs of the simulation that is then carried out. A target function (Equation 4.9) is used to evaluate the degree to which the scenario under investigation was able to fulfil the quality criteria. Through customizable weights, the multi-criteria objective function both normalises different feedback parameters – i.e. costs, delays and amounts of energy –, so that they are on a comparable scale, and prioritises the part goals. A generalized target function, based on Sobottka et al. [162], is given in Equation 4.9. In this equation, the weights (ω_j) determine the relative importance of different part-goals (k_j) towards the overall scenario fitness (f). A genetic optimisation algorithm is then used to find new values for the variables initially defined as degrees of freedom. They, in turn, are fed back to the simulation, and the process starts again. This happens until a defined stop criterion is fulfilled.

$$f = \sum_{j}^{n} \omega_j \cdot k_j \tag{4.9}$$

Following the general optimisation approach, the models representing the energy central were used to predict the performance of the overall system for representative example scenarios. These scenarios each represented a day. For this period, the overall cold ($\dot{Q}_{cold,total}$) and heat power ($\dot{Q}_{heat,total}$) demand was known and used as an input for the simulation runs. As an optimisation algorithm, a genetic algorithm was chosen. Before each run was started, a candidate operation strategy (represented as a priority vector $\vec{p}$ was generated according to some boundary conditions. This candidate strategy was evaluated through simulation. The results of the simulation run were then aggregated using a target function resulting in a fitness value. This process was repeated until a stop criterion was reached. Different target functions were used to evaluate the saving potentials for different scenarios.

As initially stated, one goal of the BaMa-project was to provide industrial companies with a tool that aids their decision making processes by giving them an insight to the effects of their actions on energy demand and emissions. Ultimately, a solution had to be developed that derives concrete operation strategies that can be put in practice by the responsible managers. In order to realise this, the method of optimisation was chosen.

Degrees of Freedom and Boundary Conditions

As was already mentioned in Section 4.1, to describe an operation strategy, the priority vector $\vec{p}$ was used. The entries of the vector are integers and represent the chillers. The position of each chiller determines its "priority", i.e. the chiller that is at the first position will be switched on first, then the second and so on. If the power demand decreases, and chillers are to be switched off, this happens in reverse order. An example vector is given here:

$\vec{p} = (8,2,4,9,6,7,5,3,1)$

The vector can be interpreted as follows: machine 8 is switched on first, then machine 2, then machine 4 and so on. Every entry in the priority vector is a unique, natural number in the range from 1 to 9. To make sure, that only vectors compliant with these rules are simulated (to reduce the search space), some boundary conditions were required. Only candidate vectors conform to these conditions were considered valid solutions and therefore used for simulation. Throughout the optimisation process, the target of the optimisation algorithm was to vary the priority vector, until an optimum was reached.

- all entries of $\vec{p}$ are unique
- all entries of $\vec{p}$ have to be from $\{1 \leq x \leq 9\}$

Even after those boundary conditions were introduced, the size of the search space, which can be calculated using $P_n = n!$, were n is the number of chillers, was rather vast. For the presented use case, which had $n = 9$, this results in $3.6 \cdot 10^5$ candidate vectors. For more complicated scenarios (i.e. such scenarios with more chillers), this number would increase dramatically. With a simulation runtime in the order of minutes per candidate on a standard PC, it would not have been feasible to evaluate all of these potential solutions, which led to the decision to employ a genetic optimisation algorithm.

Furthermore, the requirement that chillers must only be operated in *PLR* areas that were found in historical data was formulated. This increases not only the prediction quality in respect to the expected error (as it prevents extrapolation), but also reflects constraints present in the real system.

Target Function

For all valid candidate vectors, a simulation run was conducted. To make it possible to evaluate those runs, the results had to be aggregated using a target function. The fitness value (f) (4.10) was calculated based on the values of results from the simulation run. The parts of the function represent different aspects that need to be considered throughout the evaluation of a scenario. These are overall electrical energy demand (f_1) (4.11a), heat demand from external sources (f_2) (4.12a). In general, lower fitness values are preferable to higher ones. For all parts, weight factors (ω_1 and ω_2) were used to consider the specific relative priorities of the respective part goal.

$$f = f_1 + f_2 \tag{4.10}$$

In the first part of the fitness function, the total electrical power demand of all chillers is integrated for the duration of the scenario to result in the total energy demand to operate the chillers (4.11b).

$$f_1 = \omega_1 \cdot E_{el} \tag{4.11a}$$

$$E_{el} = \int_{t_{start}}^{t_{end}} \sum_{i=1}^{n} P_{el}^{i} dt \tag{4.11b}$$

The second part of the target function is used to take the total amount of heat energy that has to be supplied by external sources ($Q_{heat,net}$) (4.12b) into consideration. This, however, is only necessary whenever the total heat power provided by all chillers through heat recovery ($\dot{Q}_{wrg}$), (4.12d) is lower than the heat power that is required ($\dot{Q}_{heat,total}$) at any given point in time (4.12c). Based on the operation strategy, more or less heat is generated by chillers with heat recovery.

$$f_2 = \omega_2 \cdot Q_{heat} \tag{4.12a}$$

$$Q_{heat,net} = \int_{t_{start}}^{t_{end}} \dot{Q}_{heat,net} dt \tag{4.12b}$$

$$\dot{Q}_{heat,net} = \begin{cases} \dot{Q}_{heat,total} - \dot{Q}_{wrg}, & \text{if } \dot{Q}_{heat,total} > \dot{Q}_{wrg} \\ 0, & \text{otherwise} \end{cases} \tag{4.12c}$$

$$\dot{Q}_{wrg} = \sum_{i=1}^{n} \dot{Q}_{wrg}^{i} \tag{4.12d}$$

4.3.2 Optimisation Results

In this section, the optimisation results will be presented. In order to acquire them, a representative one-day scenario, for two different Part load ratio (PLR)s was chosen. The PLR influences the potential optimisation results since lower demand results in more possible combinations of chillers, which ultimately results in a better optimisation result. Scenarios were taken from historical records and chosen to be representative of the respective class of demand (high and low). They were defined by two time series representing the total demand for thermal cold ($\dot{Q}_{cold,total}$) and heat power ($\dot{Q}_{heat,total}$). Based on these two inputs, an optimal operation scenario is determined by the optimisation algorithm described above.

For these two base scenarios, the weight factors were varied. First, both scenarios were optimised only considering the total electrical energy demand per day for a given operation strategy (scenario 1 and 2). Then, by also setting the weight factor $\omega_2 = 1$, both energy forms, electrical energy and heat energy had to be considered by the optimisation algorithm, which led to different results. An overview of the scenarios and their respective performance improvements can be seen in Table 4.2. A reduction can be achieved with both workload conditions. As initially stated, a higher workload leads to less optimisation potential.

Table 4.2 Optimisation results for different scenarios

Scenario	PLR [%]	ω_1	ω_2	Energy Use Baseline [kWh]	Energy Use Optimised [kWh]	Reduction [%]
Scenario 1	84.4	1	0	104592	89038	14.87
Scenario 2	35.9	1	0	53958	35094	34.96
Scenario 3	84.4	1	1	115672	90295	21.94
Scenario 4	35.9	1	1	70351	41473	41.05

Even though the model error was relatively high, still even in the worst case scenario, the resulting savings would have been significant. If the results, that were acquired based on simulation models could also be achieved in reality, remains to be seen. A more elaborated parametrisation approach, which reduces model errors even more, was described by Mörzinger et al. [129]. Based on these results, which were achieved using the same framework and data from the same use case, in scenario 2, the optimal operation strategy found by the genetic algorithm uses the chillers 4, 9, 10 and 11 with a capacity-weighted model error of 13.19%. This leads to energy savings of at least 21.77% for the chosen scenario with a PLR of 35.9%. For this scenario, this is equivalent to energy savings of approximately 4770 MWh per year. If the results achieved by the implemented system, however, live up to the high expectations, an implementation of such a system on a broader scale would, therefore, be advisable.

4.4 Process Analysis

Although results from the described approach were very encouraging, the total time spent (approximately 6000 working hours) on development was high. A large team of interdisciplinary experts was required to create and implement the system. Consequently, even though the developed solution could in principle be applied to several other sites with almost no hardware costs, it seemed unrealistic that this would happen. In this section, the analysis of time recordings from employees that worked on the particular use case is presented. The goal of this analysis was to identify the major bottleneck (if it exists) that determines the necessary time. As stated in Chapter 1, the hypothesis is, that *data access* plays a crucial role (Hypothesis 1).

The data that was used for this analysis are time records that originally were created for reporting reasons due to the specifications of the funding agency (FFG) of BaMa. According to those specifications, every employee that participated in the project had to keep a record of the conducted work, the working hours and the respective working package according to the project plan. The conducted work is described in natural language and does not follow any formal regulations. An example can be seen in Table 4.3.

Relevant records for the use case under consideration came from two organisations, namely Institute for Production Engineering and Photonic Technologies (IFT) and INF. In total, 19 employees conducted a total of 23,318 work hours (t_{total}) during the project. 13 employees were affiliated with

Table 4.3 Excerpt from a time record for a single employee

Date	Work package	Description	Hours
08.01.2017	5	RKW Modellierung	6.25
09.01.2017	1	Update Projektplan	8.25
10.01.2017	6	Berndorf Band Datenanalyse	10
11.01.2017	6	Berndorf Band Datenanalyse	11.75
13.01.2017	6	Berndorf Band Datenanalyse	6.75
17.01.2017	6	Gebäudemodelle Feedback	9.25
18.01.2017	6	Gebäudemodelle Feedback	8.75

IFT and 6 with INF). Detailed records (task descriptions on a day-to-day basis), however, were only available for those from TU Wien. For INF, only aggregated values per employee and work package could be used.

4.4.1 Process Abstraction

In order to be able to analyse the time spent to create the presented results for the use case under consideration, a process model is required. Already in Chapter 1, such a model was introduced. The data science process model defines three phases that happen consecutively.

At the beginning of this process stands *Problem Formulation*. A question formulated by a domain expert. In this case, questions came from IFT. Based on formulated requirements, appropriate simulation models were chosen. Based on those models and the physical structure of the plant under consideration, they requested data from INF in the form of questions. In the presented use case, for example, the following questions had to be answered:

- What temperatures were measured at the interface of chillers of a given type?
- How many electrical power measurements are stored in connection with milling machines?
- What kind of production processes use a specific machine? What are the typical power profiles for those processes?

These questions are the input for the second step, *Access*, which covers all measures necessary to generate the dataset needed to answer the formulated expert question. Based on these questions from IFT, the contact person at INF had to identify the engineers responsible for all relevant subsystems. For complex questions, several engineers had to be consulted, before the questions could be translated to queries to the data storage. As it is the case in most industrial applications, for the considered use case, RDBMS were used for storing that data. Therefore, the task of accessing the data sources has to be done by IT experts which then, in turn, formulate custom SQL queries based on their perceived understanding of the questions and the knowledge about the schemas of the relevant data storages.

The last step, *Analysis* covers all necessary steps to process the data. Depending on the initial question, different methods can be used to find answers. In this instance of the process, datasets

produced by INF were handed over to IFT. These datasets were then transformed, processed and finally analysed in order to retrieve the parameters for the simulation models.

4.4.2 Analysis of Time- records

Time records do cover all work that was conducted within the project (t_{total}). However, only a part of that work is relevant for the use case under consideration and falls into the categories of the process model. Therefore the following steps were carried out to filter irrelevant records. Resulting working hours per organisation after each filter step can be seen in Table 4.4.

1. For both IFT and INF, time records were filtered based on the role of the employee that carried out the task. Only tasks carried out by those employees that held the role of engineers or scientists were considered for further analysis. Others, such as administrative personnel or managers were excluded — the remaining working hours after this step were summed up as t_{empl}^{IFT} and t_{empl}^{INF}, respectively.

2. Then, again for both IFT and INF, tasks that were assigned to work packages that are not directly related to the data science process (i.e. dissemination or project management) were excluded — the remaining working hours after this step was summed up as t_{WP}^{IFT} and t_{WP}^{INF}, respectively.

3. In the case of IFT, time records had to be excluded that did not seem to concern work that was not dedicated to any of the use cases provided by INF. This step was skipped for INF, as any of the works that were carried out by their employees, by definition, were concerned with use cases from INF. Any tasks carried out by INF fulfilled this requirement — the working hours remaining after this step were summed up as t_{INF}^{IFT} and t_{INF}^{INF}, respectively.

4. Throughout the project, two different use cases were developed in connection with INF. Only one of them is relevant for this analysis, which is why another filter stage was necessary. Based on the natural language description of tasks, those that were not connected with the use case under consideration were excluded manually. The main issue at this stage was that detailed records including these descriptions were only available for IFT. For INF, however, the assumption was used that the time they spent on each of their use cases was distributed in the same share as it was for IFT- see (4.13a) — the working hours remaining after this step were summed up as t_{UC1}^{IFT} and t_{UC1}^{INF}, respectively.

5. In the final step, those tasks that were not directly concerned with the data science process (concerning, i.e. implementation into existing systems or optimisation) had to be selected manually. Again, for this step, the task description was used in the case of IFT. For INF, a similar approach as in the previous step was used. The working hours remaining after this step were summed up as t_{KDD}^{IFT} and t_{KDD}^{INF}, respectively- see (4.13b).

Table 4.4 Total working hours per organisation after applied filtering was applied

	t_{total}	t_{empl}	t_{WP}	t_{INF}	t_{UC1}	t_{KDD}
IFT	15561	8532	6034	2757	2113	1817
INF	7757	7229	5355	5355	4105	3530
Sum	23318	15761	11389	8112	6219	5348

6. In the final step, the remaining tasks are classified. It is assumed, that all time that was spent by INF can be classified as *Access*. In the case of IFT, the tasks have to be classified as either *Problem Formulation* or *Analysis*. The working hours remaining after this step were summed up as t_{Form}, t_{Access} and $t_{Analysis}$, respectively. As a general rule, any task description mentioning data analysis" or "parametrisation" was considered to be *Analysis*. The rest (mainly described as "modelling") was classified to be *Problem Formulation*.

$$t_{UC1}^{INF} \approx t_{INF}^{INF} \cdot \frac{t_{UC1}^{IFT}}{t_{INF}^{IFT}} \tag{4.13a}$$

$$t_{KDD}^{INF} \approx t_{UC1}^{INF} \cdot \frac{t_{KDD}^{IFT}}{t_{UC1}^{IFT}} \tag{4.13b}$$

The conducted analysis, due to the quality of available data, has several sources of uncertainty that need to be discussed and pointed out. First, there are the task descriptions, which do leave some room for interpretation and often it is not apparent how to classify specific tasks. Combined with knowledge regarding the overall project and respective periods in which certain use cases were worked on, the descriptions can be classified with a sufficient amount of accuracy. Furthermore, and probably most importantly, detailed task descriptions were only available for one of the two organisations that worked on the discussed use case. Consequently, it had to be assumed that the timeshare that was spent per use case, and within the use case for data related tasks, was the same as it was for IFT. Finally, it was assumed that any tasks that were made on the side of INF could be classified as *Access*. Given the way the tasks within the use case development were shared between the organisations, this seems reasonable. If anything, the notion that no *Access*- tasks had to be carried out on the side of IFT leads to an underestimation of the time that was spent on this step. Often, datasets could only be handed over in the form of database-dumps, which led to a considerable effort on the side of IFT that might also be considered to be related to *Access* instead of *Analysis*.

4.4.3 Result Interpretation

The result of the analysis can be seen in Table 4.5. As it seems, indeed the largest part of the total working hours was spent on *Access*. The resulting 66% are below what was reported by Kharlamov et al. [100]. Nevertheless, this is remarkable, as for the first step (*Problem Formulation*) much had to be done that would not be required outside of a research project. In this particular case, within this

Table 4.5 KDD related working hours classified according to the respective process phases

Process Phase	**abs**h	**rel**
Formulation (t_{Form})	1062	20%
Access (t_{Access})	3530	66%
Analysis ($t_{Analysis}$)	756	14%
Sum	5348	100%

phase, feasible simulation models had to be researched, tested and combined to represent the facility. This certainly increased the working hours in this phase compared to typical industrial settings.

Among the reasons for this result, and particularly the relative importance of *Access* phase, are inconsistent naming conventions, changing database schemata and communication problems between domain- and IT Experts. At one point, up to six different domain experts (not all of whom were within the project team) had to be consulted in order to collect all the necessary information to formulate a database query.

As was shown, most of the time spent in order to acquire usable models of the chillers in place was spent on the second step (*data access*). The following reasons were identified for that:

- It requires knowledge about the structure of the underlying databases and their schemas as well as the correct names of the data-points which are to be extracted. This knowledge is hardly ever in the hands of domain experts formulating the initial questions. As a consequence, large amounts of time are spent communicating with IT specialists about what kind of data is needed.
- Data is not stored in just one database. Often, additional information from various data sources had to be used in order to answer queries. Furthermore, in order to generate queries, human knowledge from several different domain experts had to be used as well.

As a consequence of this analysis, solutions to reduce the time demand required for *Access* are necessary. Therefore, in the following section, the development, implementation and evaluation of a PoC for the energy central application scenario will be described.

4.5 Proof of Concept

To address the identified limitations of the state of the art solution described above (Section 4.4), OBDA was identified as a promising technology, as it was already successfully applied to other, similar applications [100]. In order to determine if this approach would also be suitable for the scenario at hand, we built an OBDA-based PoC. The primary goal of this initiative was not to create a fully functional, production-ready solution, but rather to find out and illustrate the benefits and caveats of the technology from the application scenario perspective.

First, raw monitoring data already provided for the model parametrisation described above (Section 4.2.1) was used. The CSV- files were however stored in a RDBMS. As the original schema was not

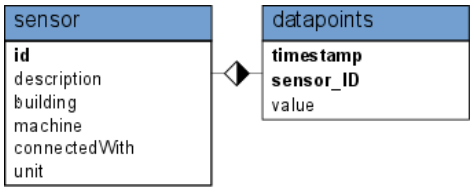


Figure 4.7 PoC database structure.

known, the form depicted in Figure 4.7 was used. In this example, relevant information regarding the measurements is included in the SENSOR table. In this table, necessary information on the unit of measurement of the respective sensor (unit) and connections with other sensors (connectedWith) are stored along with other data.

4.5.1 Ontology Development

As the primary goal was to evaluate the general technical applicability and motivate further development, it was necessary to build an ontology which illustrate the features Semantic Web technologies provide, while showing how a concrete example use case in the respective field could look like.

Example expert questions were formulated at the start of the development process. They were inspired by requests, which had to be answered throughout the parameter estimation process. Some example questions are:

- Which chillers are located in building XY?
- Show all chillers located in country YW.
- Which chillers are connected with recooling plant YZ?
- What was the electrical power demand of all chillers with heat recovery between 01.03.2017-01.04.2017?

Based on those questions, and with the requirements formulated above, two ontologies were created: the *Top Level Ontology* and the *Chiller Ontology*. Even though these ontologies could have also been combined, the decision to split them was taken to illustrate the fact that ontologies can be used to combine conceptualisations from several domains.

The Top Level ontology, depicted in Figure 4.8a, represents the main domain concepts and their relations. Through this ontology, more detailed ontologies such as the one depicted in Figure 4.8b could be combined. The concept MACHINE from the top-level ontology is further specialised in the Chiller ontology concerning various chiller types. Other subclasses of machines could, for example, be production machines, but also logistics equipment. Another way to expand the ontology might be to describe the building domain. This, however, was not in the scope of the PoC.

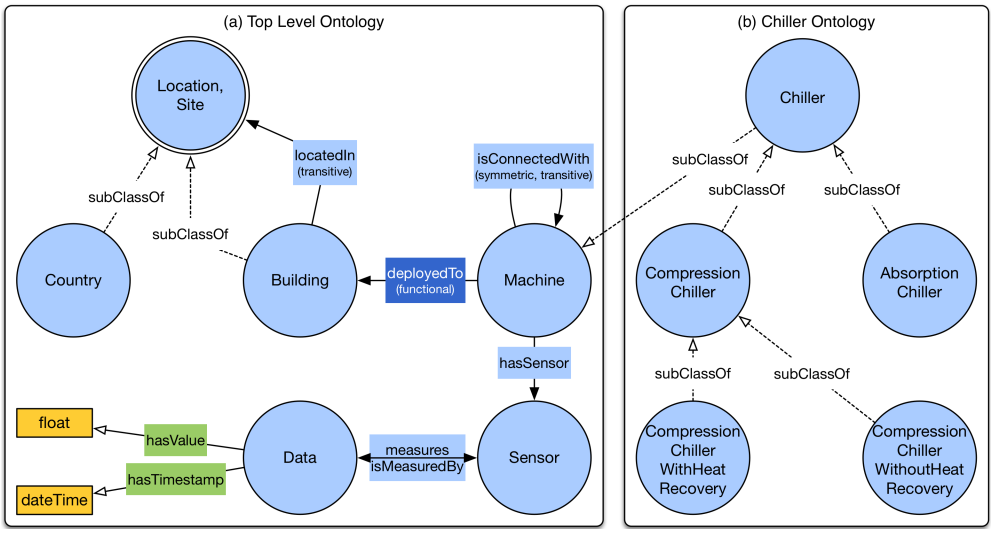


Figure 4.8 Top Level and Chiller Ontologies [128].

4.5.2 Mappings

A mapping combines results from SQL queries with ontology elements such as concepts and predicates. This can be seen as a virtual Tbox. Mappings are then used to translate SPARQL queries (constructed with terms from the ontology) into SQL queries (executed on the underlying databases). Ontop's Protégé plugin was used to implement the PoC. This plugin aids the mapping process. In Figure 4.9, one of those mappings is depicted. In this example, targets from the ontology (DATA, which is a concept, or ISMEASUREDBY, which is an object property) of the ontology are linked with the source database. This link is established through a SQL query. The columns of the resulting table (in this case the DATAPOINTS- table) are linked with the respective elements of the ontology. Specifically, a datapoint ID (dpid) is generated for each sensor and each timestamp. The result of this is used as a URI of a instance of the concept DATA. Furthermore, this instance is connected with timestamps and values through the predicates HASTIMESTAMP and HASVALUE as well as with a Unique Ressource Identifyer (URI) representing instances of the concept sensor used to generate the data point.

4.5.3 Query Formulation

In this section, two queries are presented which are based on questions from domain experts by making use of the terminology of ontologies. The first query (Listing 4.1) returns time series data for a specific SENSORID in a defined period. If this information (SENSORID and period) is at hand, already a traditional database powered system can be used to access data. This, however, normally is not the case. To identify the correct SENSORID, different sources of information such as factory building plans, connection diagrams, and wiring schemes have to be consulted. Those sources might or might not be digital, but rarely are machine-readable. This leads to the problem, that, even if all sources are available and identified (which also rarely is the case), the task of SENSORID identification has to be

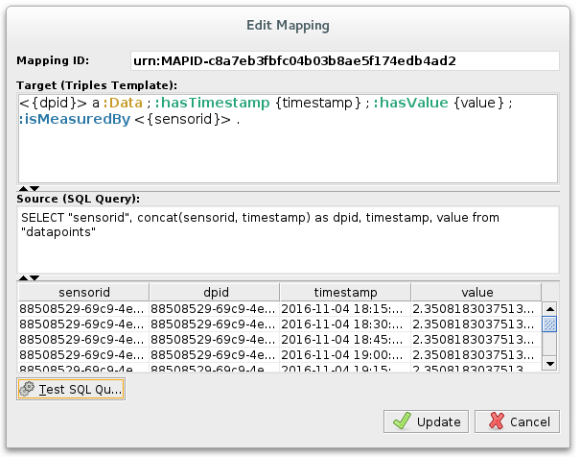


Figure 4.9 Example mapping in Protégé editor.

```
PREFIX : <http://ontologies.ift.at/INF_Top_Level.ttl#>
SELECT ?Timestamp ?Value
WHERE {
?Data :isMeasuredBy ?Sensor ;
:hasTimestamp ?Timestamp;
:hasValue ?Value .
?Sensor :name ?SensorName .
FILTER (?SensorName = "c343144f")
FILTER(?Value > "2721").
FILTER(?Timestamp >= "2016-10-12 00:00").
FILTER(?Timestamp <= "2016-10-13 00:00").
}
ORDER BY ASC(?Timestamp)
```

Listing 4.1 First query

done manually which makes it very costly. Furthermore, as the information sources are scattered and under the responsibility of different departments, they tend to be inconsistent or out-dated.

In the second example query (Listing 4.2), the technological capabilities of Semantic Web are used to identify sensors based on example knowledge such as machine types and machine connections. This information, without the use of knowledge graphs, would have to be extracted from factory plans or wiring diagrams. Here, only data generated by sensors deployed to machines which are connected to other machines is returned. The results were filtered to limit the results to those, where the second machine was one particular cooling tower (identified by its name). These sensor names could then, in turn, be used to query the respective time series data. To illustrate reasoning, only the predicate ISMEASUREDBY was explicitly mapped to data from the underlying dataset. Nevertheless, the second query makes use of the MEASURES predicate (the inverse of ISMEASUREDBY).

```
01 | PREFIX : <http://ontologies.ift.at/INF_Top_Level.ttl#>
02 | SELECT DISTINCT ?Machine ?SensorName ?Data
03 | WHERE {
04 | ?Machine :hasSensor ?Sensor ;
05 | a ch:Chiller ;
06 | :isConnectedWith ?otherMachine.
07 | ?otherMachine :Mname ?otherMachineName .
08 | ?Sensor :measures ?Data;
09 | :name ?SensorName .
10 | FILTER (?otherMachineName="RKW3")
11 | }
```

Listing 4.2 Second query

4.6 PoC Evaluation

In order to evaluate the PoC, semi-structured interviews with stakeholders of the project from INF were conducted. The proposed approach is generally seen positively by the involved stakeholders, as can be judged from the interview results. Therefore, the hypothesis that virtual knowledge graphs can improve data access for domain experts in this application seems plausible. In order to be truly useful for users and therefore facilitate the adoption of simulation-based approaches to optimisation tasks, however, further improvements are necessary.

4.6.1 User Feedback

A preliminary evaluation was carried out with the presented PoC. The two main goals of the evaluation were to determine how potential users would interact with an OBDA-based system such as the proposed one and to identify potential barriers to large-scale adoption of such a system within the company. In total, four semi-structured interviews were conducted with:

- Two Users. Potential users of the proposed system would be Energy managers, who are responsible for the optimisation of energy systems both concerning energy demand and system reliability. This involves monitoring and improving the factory.
- One IT Application Expert. The IT application experts role in the context of this thesis is to deliver domain experts (such as energy managers) with data sets based on their specifications.
- One Global Head of Facility Management. His role, in the context of this thesis, is to globally evaluate measures and make them comparable, so that best practices are shared as fast as possible.

Each of the participants had a particular role within the project and therefore a different perspective on the presented prototype. This makes it possible to evaluate the prototype not just from the user perspective, but also from the perspective of IT experts and the global facility management. This also unveiled additional application scenarios for this technology. After a short reminder of the

functionality of the prototype, the interview was started. The following questionnaire was designed to guide through the interview. Some questions were skipped if they had already been clarified by previous answers.

1. When thinking about the chiller use case, do you think virtual knowledge graphs would have improved our work?
2. What are the main challenges in your current field of work (generally)?
3. What kind of other applications for an OBDA system can you imagine in your current field?
4. If you had to choose one feature that developers should focus on in order to make the OBDA system more useful, what would that be?
5. If you had to use a OBDA system regularly, what requirements should it fulfil?
6. Are there any circumstances that would make the OBDA system unusable for you?
7. What are your main concerns regarding the implementation of an OBDA system in your organisation?

User Perspective

In retrospective (*question 1*), the point was made that it is expected that the OBDA-based prototype could have significantly reduced the amount of time spent on data acquisition compared to the approach where a state-of-the-art approach was used. The main factor, according to the users, is the fact that domain experts would have been able to access the data directly. At present, at the local level, data is currently stored in different software solutions, and the combination of data from those datasets (even though it is often necessary) poses a big challenge. Also, semantically stored metadata would reduce the necessary coordination efforts between different domains drastically.

In this context, the current challenges (*question 2*) become evident. Currently, data storage is based on a collection of different, heterogeneous solutions which hardly interact with each other. Energy management, however, is an inherently interdisciplinary field which requires not only cooperation between experts from different fields, but also a seamless connection of the respective software tools and the associated data.

Regarding usability and further features (*questions 4-6*), the main requirement would be a graphical user interface for the current prototype. Browser-based solutions with simple elements such as drop-down menus and single line query tools would be solutions. As a positive example of how such a system might look like, current solutions such as LodLive[3] [43] were named. Apart from this, system stability, speed and system support were expressed as crucial requirements.

[3] http://en.lodlive.it/

The main concern from the user side (*question 7*) is that concrete application scenarios need to be presented, before the system can be implemented. This is, however, the prerequisite for the acquisition of concrete results.

IT- Experts

In the past project (*question 1*), processing requests from the domain experts took less time then it would typically have. This is because in this particular case the IT expert had a background in facility and energy management. Still, misunderstandings surfaced, and requests had to be reiterated several times. Therefore, the expectation that the Semantic Web could have reduced the time demand for data access requests significantly was confirmed from the IT side as well.

From the IT side, data integrity and quality are among the main challenges (*question 2*). A high number of sensors is connected to different storage systems. Sensors might sometimes stop generating or transmitting data. This is currently checked via monitoring of the storage demand. Through this, the data quantity is checked but not the quality. In the current system, only human supervisors can qualitatively identify suspicious measurements. This is not feasible, however, which leads to potentially wrong measurements. If usability (*question 4-6*) was increased through a user interface, a system such as the one proposed would make it possible to delegate the task of data access to the respective domain experts. This would reduce the cost of data analysis significantly. Again, the requirement for system stability was emphasised. Regarding potential challenges from the IT Experts side (*question 7*), security and data governance would need to be adapted and taken into consideration if such a system was to be put into operation.

Management

The statements reported in the previous sections (*question 1*) are also reflected in those of the management. The data access challenge in the previous project was far greater than anticipated. This is the main aspect that Semantic Web technologies could have addressed. The primary challenge (*question 2*) for the Management is the task of centrally accessing data not only from different software systems within one site but globally across sites. This makes it necessary to be able to combine data from different data storage, mediating different spoken languages and also entirely different conceptualisations of the problem domain — conventional approaches to harmonising data models and infrastructure are always at risk of becoming outdated. When asked about implementation challenges (*question 7*), additionally to the already made statements, the novelty of the proposed approach was mentioned. Resources are already very limited within the IT department due to "daily business". Therefore, a significant amount of work would need to be done to prove the usability of a Semantic Web-based prototype. Potential adoption barriers for such a system might, according to the participants, primarily arise due to the "novelty" of the proposed technology and a general lack of IT skills for managing those. Also, in order to illustrate the usefulness of the proposed solution, the

resulting tool would need to have a certain minimum size, and therefore, the initial implementation effort is rather high.

4.6.2 Necessary Features

Semantic Web technologies seem to be feasible to solve these problems. Especially OBDA seems to be a promising option. Based on feedback from the stakeholders of the BaMa- project, and through insights gained while developing the PoC, the following features were identified as necessary for the intended application scenario. The most apparent additionally required feature that can be derived from the interviews is the request for an appealing and stable user interface. This is a matter of engineering works, as the general capability of OBDA systems to create and answer queries efficiently also for large datasets was already shown by Soylu et al. [165, 164, 163].

Data Integration

From the interviews, it became clear, that the integration of different data sources is an important feature. The possibility to use an ontology to access all relevant data sources without requiring detailed information about the underlying structure was frequently mentioned as one of the main benefits of the proposed solution. This leads to the topic of OBDI. The presented PoC and the subsequent discussion, especially with the management also showed, that the same data can be seen and processed differently, depending on the role of the user. Currently, this is done via respective queries on the application level. Alternatively, ontologies could be used and tailored so that different domains and their specific concepts would be described. Those ontologies could be linked via ontology federation tools that create links between similar or even equivalent concepts from different ontologies. This would result in a reduced mapping effort, and the data could stay in one place.

Temporal Concepts

A second requirement can be derived from the nature of the models in question. The processes happening within the modelled machines are temporal by nature. This also becomes clear from the first presented query. Many concepts can only be defined if expressions such as "coincidental" or "subsequent" can be used. With this ability, the queries would become much more straightforward. To harness this, any further development should incorporate this kind of concepts. The underlying theory is described in 3.3.

Chapter 5

Application Scenario: Manufacturing Rig

The database that was used in the previous scenario was significantly simpler than the actual one in place at Infineon. Without real-world restrictions, requirements towards the used database- schemas and the underlying system structure can only be guessed which might lead to illegitimate simplifications. This introduces a chance for potential bias and therefore limits the validity of any result derived from experiments on a system that builds on it. Therefore, a second application scenario is introduced in this section, which will be used for the development of further OBDA prototypes.

5.1 Scenario Description

The second application scenario is concerned with the experimental investigation of Vibration Assisted Drilling (VAD). These type of processes are a special kind of drilling processes, where axial feed movement is overlain by axial oscillatory movements. This is done to improve the formation of chips. Oscillation frequencies can range between 500Hz-16kHz, amplitudes are in the order of μm [141]. It has been successfully applied to materials such as Titanium [135], Aluminium [91], CFRP [19], and nickel-based alloys [161].

In order to investigate this technology, a dedicated manufacturing rig was used. The goal of this investigation is to test process parameters for different tools and materials and find combinations that are optimal due to some defined criteria. Apart from a newly developed machining spindle, more than 50 sensors (measuring for example forces, accelerations, temperatures or sound) are connected with the manufacturing rig. The general structure of this rig, as well as the deployed sensors, will be described in Section 5.2. The core process schema is depicted in Figure 5.1.

The intended application area of the results generated through the experiments is the aviation industry. Consequently, not only a focus has to be laid on the quality of the machined features, but also specific materials are used. Apart from those already mentioned, potentially arbitrary combinations of them in the form of compound plates were investigated.

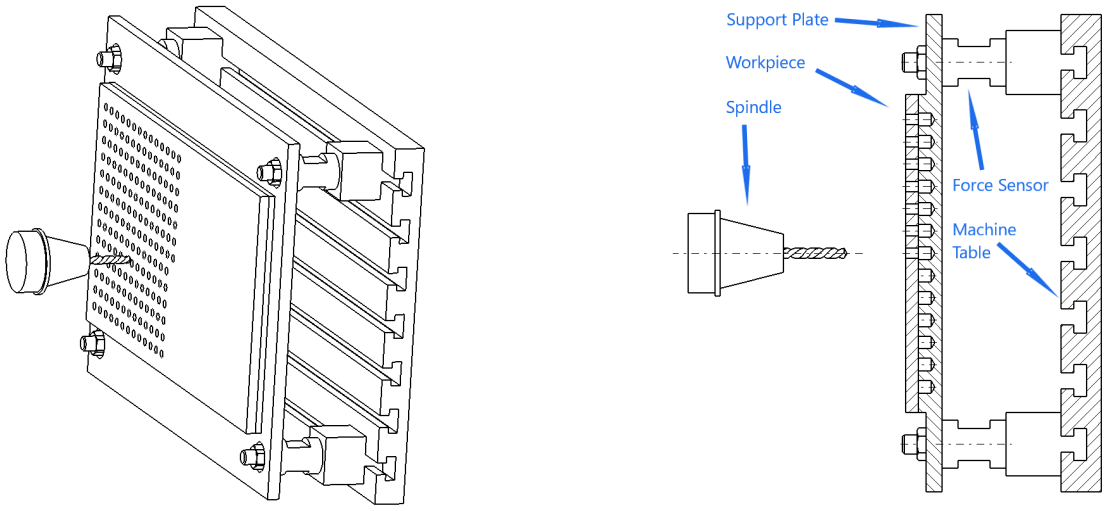


Figure 5.1 Schematic depiction of the VAD drilling process

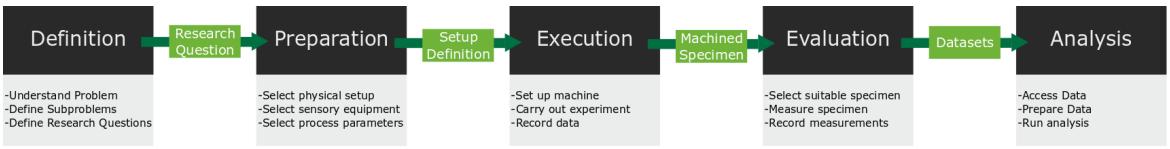


Figure 5.2 Process for experimental research in manufacturing

The general structure of the conducted experiments is relatively straight forward. For any combination of materials, tool and process parameters (experimental parameters), a set of holes is drilled. Data generated by sensors throughout this process are stored for subsequent analysis. Furthermore, quality criteria such as diameter, roundness or surface roughness are measured afterwards and stored along with the sensor data and the respective choice of experimental parameters. This data can then be used for analysis to find, for example, correlations between specific sensor readings and quality measures.

To facilitate the structured execution of experiments, a process (Figure 5.2) was defined. This process is described both generally and for the application scenario at hand in the following section. It was defined with experiments in discrete manufacturing in mind, but might also be applicable for a broader range of application scenarios. At the beginning of this process stands the definition of the endeavour. Motivated by developments such as novel materials, product features (their physical shapes) or tools, problems arise. Some of these motivate the formulation of research questions and are therefore subject to research. Different methods are at the hands of researchers, and those that are best suited for the particular question have to be selected.

5.1.1 Definition Phase

As already indicated, not all research questions can or rather should be answered through experimental investigation, which is a costly and hardware intensive way of research. Often, a thorough literature review or numerical simulations might provide similar, or better results than experiments. Two main criteria result in the popularity of experiments in the manufacturing domain. First, technological phenomena that appear when actual manufacturing processes such as milling, drilling or turning happen, are often too complicated to be covered by simulation models. Furthermore, the potential success of literature reviews is often limited since relevant data, and scientific results are often not available to researchers because certain problems have not been investigated yet or that results are not publicly available.

Consequently, the first step of (any) research process, is the formulation of research questions. In the application scenario at hand, example questions of the following types were asked:

- "What is the average burr height if CFRP is machined with tool XY?"
- "How do process parameters need to be chosen in order to maximise the tool lifetime for tool X when machining material Y?"
- "Can process force measurements be used to predict the surface roughness of a drilled hole?"

Those questions from the DEFINITION- phase that are identified as suitable for experimental investigation, are the input for the PREPARATION- phase.

5.1.2 Preparation Phase

In this phase, research questions are used to choose a suitable machine and its setup as well as tools and raw materials (physical setup) that will be used in the course of the experiment. Furthermore, the necessary sensors and their parametrisation need to be specified in this phase along with the respective machining processes, which are defined through process parameters.

In the case at hand, the research question leads to a set of experimental parameters which will be used. Also, parameters concerned with auxiliary systems such as suction and cooling system are defined in this step. The decisions made are recorded manually by researchers, using predefined spreadsheets. Furthermore, the chosen process parameters are used to generate NC-programs for the manufacturing rig automatically. Lastly, a unique key is generated which makes it possible to identify the experiment and therefore keep track of the context of the data that is generated throughout the actual experiments. Based on the specification created, the manufacturing rig is prepared, which might include connection of additional/ removal of excess sensors and physically mounting tools and raw material potentially along with special appliances such as fixtures or cameras. Once the setup is finished, the machining processes are started.

5.1.3 Execution Phase

Based on the prepared manufacturing rig, manufacturing processes can be carried out according to the defined specifications. If all processes finish as intended, the experiment is concluded. Experiments however might, due to different reasons, have unexpected outcomes which might include tool breakage or malfunctioning software. In that case, again, the generated data has to be flagged, and the respective processes have to be repeated with (potentially) different process parameters.

For the application scenario at hand, specimen (plates) are mounted on a dedicated fixture. Once the correct tool is mounted, and the previously generated NC-program is available on the machine, the setup is concluded by manually entering the experiment identifier code at the machine's terminal. This code is used to link sensory data generated throughout the experiment with the respective meta-information. Sensors in this respect are devices that generate time series- value pairs which represent measurements of physical quantities such as force, electrical power, temperature, acceleration, velocity or position. Those quantities can be interpreted differently based on the exact position of the sensor within the machine. Data generated by each sensor in this phase primarily has the form of time series. As such, the data is annotated, transferred and stored.

5.1.4 Evaluation Phase

After all processes are concluded, the process results (i.e. machined specimen) are EVALUATED. If necessary, they are filtered, and only those that are identified as suitable for further processing are selected for investigation. In this particular case, this involves the measurement of geometrical features. Potentially, after evaluations, some parts of the experiment need to be repeated.

In this application scenario, the following quality criteria of the created boreholes are measured. The exact approach to determining them is described in Section 5.2.2.

- Surface roughness: The roughness of the manufactured surface.
- Borehole roundness: A measure for the deviation of the perfectly circular shape of a respective borehole
- Burr height: Both at the entry and the exit of any borehole, potentially sharp edges can build up. Those are referred to as "burrs."
- Chip residues: Depending on the material and process parameters, chip residues can remain within the borehole after the manufacturing process.
- Delamination: In the case of CFRP, delamination can occur both on the entry and the exit of every borehole.

Again, similar to meta information concerned with the specification of experiments, in contrast to the data that is generated by sensors, data from these quality measurements are entered manually and do not have the form of time series. Just as the time series data, however, quality data is linked with

the defined experiment through a key to making it available for the final phase, which is concerned with the analysis of generated data.

5.1.5 Analysis Phase

In the final step, generated data is ANALYSED. In this phase, data that was generated in the previous phases is accessed, rearranged and preprocessed until some analysis algorithm can be applied to it. Such algorithms could, for example, be the calculation of statistical measures such as average or standard deviation but also include more advanced methods such as training of predictive models or clustering of the data. Very often, data is processed in order to be presented graphically.

In the application scenario at hand, all experimental data is stored in a single relational database. Meta information, sensory data and quality data are stored and linked to each other through unique keys. Data is accessed using database clients, analysis tools such as MATLAB or programming languages such as Python.

5.2 Physical Setup

The manufacturing rig depicted in Figure 5.3 is used in this scenario to conduct experiments and investigate drilling processes. Even though the machine under consideration is somewhat unique, both due to the number of additional sensors and in respect to its ability to carry out VAD processes, it is still comparable for the general class of machining centres that can be found in any larger manufacturing company.

5.2.1 Experimental Rig

At its core, the experimental rig used in this application scenario is based on an adapted Hueller Hille NB-h90 machining centre. Several different sensors are deployed to the manufacturing rig as well as a special spindle which makes it possible to carry out VAD processes. The machining centre based on the manufacturing rig is a 4-axis manufacturing rig. These axes correspond to three translatory (x, y, z) axis and one rotatory (B) axis. Additionally, a suction unit to remove chips and dust from the drilling process and a cooling system were deployed to the rig.

Within the manufacturing rig, the specimen is attached to a fixture plate, which is attached vertically. These specimens are plates (150x300mm) and could potentially have several layers. Materials processed in this scenario where Aluminium, Titanium, CFRP and combinations of them. Drilling holes are positioned on the plates in a fixed grid (Figure 5.4). Specimen were identified through their unique ID.

Figure 5.3 Depiction of the Hueller Hille NB-h90 manufacturing rig.

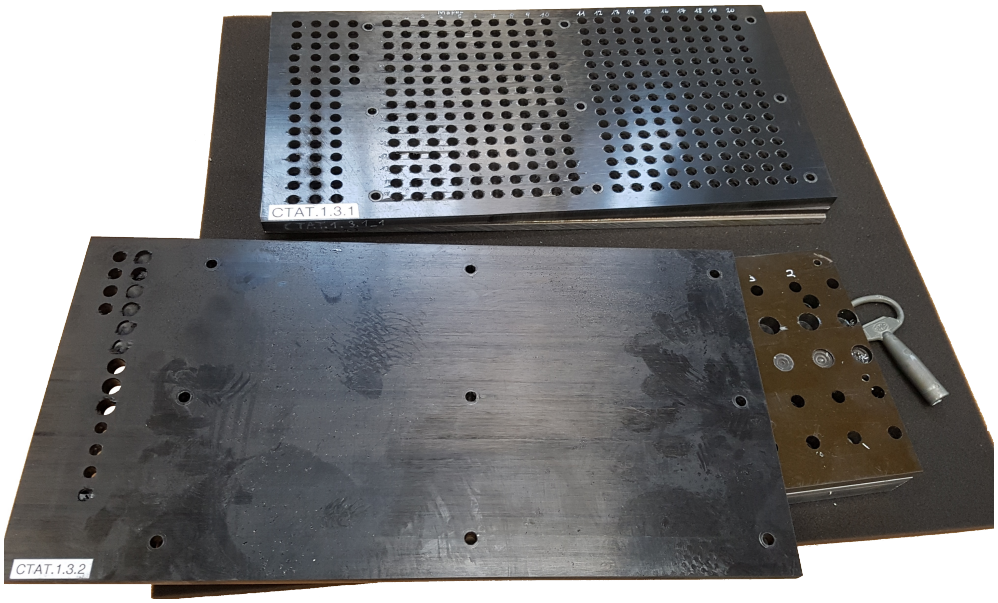


Figure 5.4 Example Specimen.

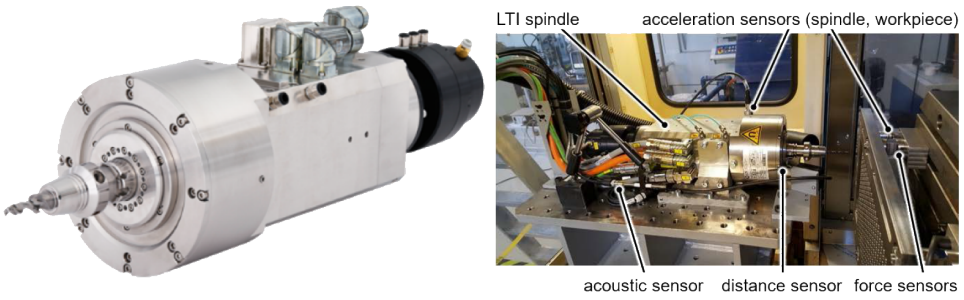


Figure 5.5 Spindle and experimental setup.

Spindle

An electromagnetic spindle system LeviSpin by LTI Motion as displayed in Figure 5.5 was used throughout the experiments. As can be seen, the machines setup is rather unconventional. The spindle, which is capable of generating axial oscillatory movements through active magnetic bearings, is mounted on the machine table. The system has a rated power of about 6.5 kW at a rotational speed of 12,000 rpm. The maximum vibration amplitude, is 0.12 mm. Both amplitude and frequency (0-300 Hz) can be set independent from the spindle speed.

Sensors

A summary of sensors that are deployed to the system can be seen in table 5.1. In total there are 52 sensors used to measure relevant quantities. As can be seen, these sensors measure a wide variety of physical quantities at sampling rates between 52 kHz and 20 Hz.

Position, velocity and acceleration of all three translatory machine axis are measured. These signals can not only be used to verify the positions of drilled boreholes but also to calculate KPIs such as material removal rate. Furthermore, for each specimen mounting point, there are sensors to determine acceleration and apparent force. Based on these signals, process stability and other quality-related investigations are possible. Electrical power intake is measured both for the machine in total and the spindle specifically. Such measurements can be used to calculate the energy intake for processes and therefore not just evaluate ecological aspects (such as CO_2 footprint), but also make assumptions regarding remaining tool lifetime. Furthermore, a microphone is deployed to the working area of the manufacturing rig. Through processing (i.e. Fourier transformation), sound signals can be analysed and provide an additional source of information on process stability. Temperature is measured at three positions, two located within the working area and one on the outside. Lastly, there are 15 other sensors deployed to the system which measures user inputs such as override or process parameters such as spindle speed. These sensors, in contrast to the others, generate event logs instead of time series logs.

Table 5.1 Deployed Sensors

Physical Quantity	Unit	No. of Sensors	Samplingrate [Hz]
Acceleration	$\frac{m}{s^2}$	7	52000
Sound	Pa	1	52000
Force	N	4	2000
Pressure	bar	2	1000
Temperature	$°C$	3	1000
Flow	$\frac{Nl}{min}$	1	1000
Position	mm	8	400
Velocity	$\frac{mm}{s}$	5	400
Electrical Current	A	3	400
others	-	15	400
Electrical Power	W	3	20

5.2.2 Quality Measurements

The quality of drilled boreholes was evaluated through a set of relevant features. For each of the features, a particular measurement technique and boundaries for acceptance were defined. The measured features are determined by the quality standards defined for the aviation industry.

Diameter and Roundness

For each borehole, the diameter is determined. For this, a MarCator 1087 BR[1] measurement device (Figure 5.6) is used. For each borehole, two separate diameter measurements with 90°offset were conducted to get intermediate results for the hole roundness. To make sure that the diameter is measured at a constant borehole depth, a spacer sleeve was used.

In a subsequent step, a Zeiss Prismo[2] coordinate measuring machine is used to determine the borehole roundness (Figure 5.7). This machine is used to generate approximately 400 spatially distributed measuring points which are then converted to a reference cycle using a least-squares algorithm. Relative to this circle, both the global minimum and maximum diameter are computed. The roundness is defined as the difference between the diameters that correspond to each of these two points (Figure 5.8). For each hole, this roundness value is determined based on three different hole depths ($t_b = [2mm, 4mm, 6mm]$).

Burr height

In order to determine burr heights, both at the inlet and the outlet of boreholes, again a MahrCator 1087 BR measurement device was used. A spacer sleeve was used to predefine the mounting height of the measuring device above the hole. To make sure that the highest possible burr height is measured, a measurement plate was used (Figure 5.9).

[1] https://www.mahr.com/en/
[2] https://www.zeiss.com/

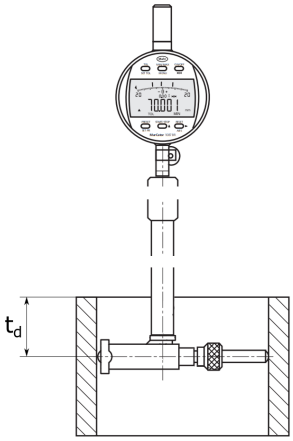


Figure 5.6 MarCator 1087 BR.

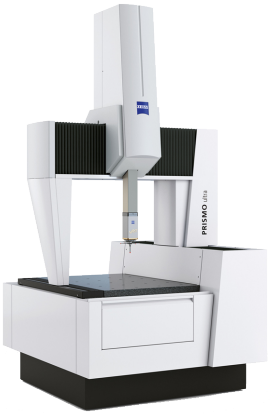


Figure 5.7 Zeiss PRISMO navigator.

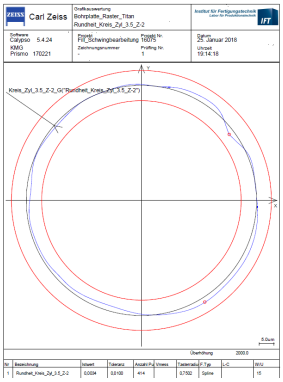


Figure 5.8 Example roundness measurement plot.

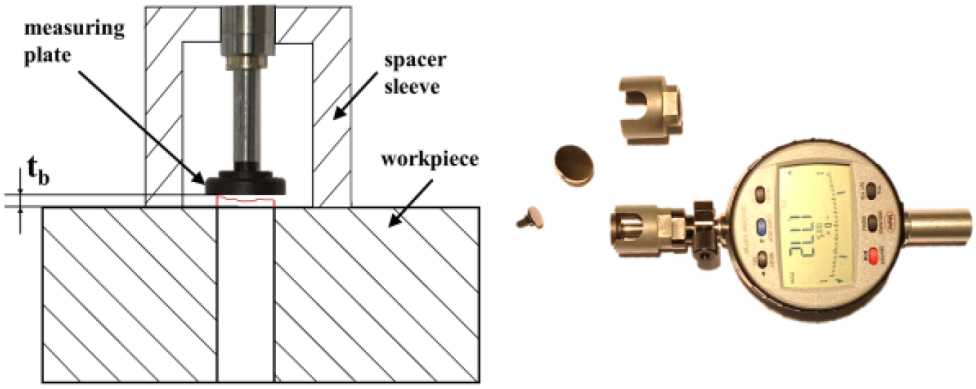


Figure 5.9 Burr height measurement principle (left) and measurement device with measurement plates (right).

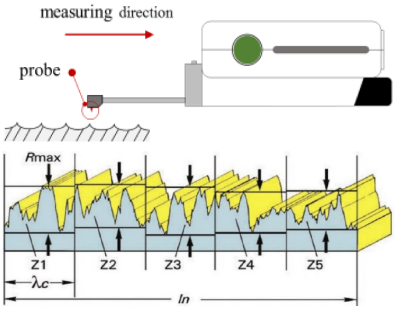


Figure 5.10 Roughness measurement principle.

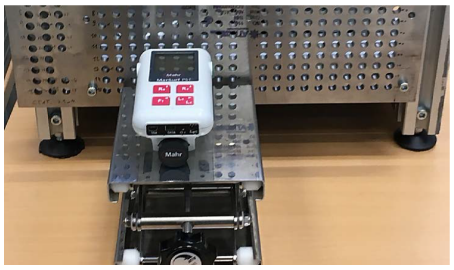


Figure 5.11 Roughness measurement set-up.

Roughness

For quantification of the surface quality, the average roughness depth R_a is determined according to DIN EN ISO 13565. Figure 5.10 shows the schematic illustration of an example surface after machining [175]. Length l_n is the minimum length that is required to carry out the measurements. It is divided into 5 subsequent sections of length λ_c. Similarly to the approach described for roundness, the difference between the highest peak and the lowest groove (R_z) is calculated. Furthermore, the average roughness depth R_a is determined through calculation of the arithmetic mean. Both of these measured are determined with the MarSurf PS10[3] roughness tester (Figure 5.11).

5.3 Data Management

As already mentioned above, three categories of data need to be stored and analysed in the applications scenario at hand:

- Experimental Metadata: Also referred to as experimental parameters, which are defined in the DEFINITION phase. This kind of data is generated manually by researchers.
- Sensory Data: time series data generated automatically by sensors such as force or temperature measurements deployed to the manufacturing rig. It has to be linked to experiments in order to be useful for later analysis.
- Quality Data: Data which is concerned with geometric features of the created boreholes such as surface roughness or hole roundness. Again, this type of data is generated manually by researchers and needs to be associated with a particular experiment similarly to sensory data.

Before it can eventually be stored data needs to be transferred to the database. The way this is done depends on the respective data category. In the case of metadata and quality data, researchers insisted on a spreadsheet tool to generate and interact with data. Therefore, standardised spreadsheet

[3] https://www.mahr.com/en

templates were created and automatically scraped for each experiment. In the case of sensory data, a National Instruments cRIO[4] industrial PC was used. Sensors were either connected with the industrial PCs I/O-module directly (i.e. microphone) or through the machines numerical control (i.e. position sensors). The industrial PC then collected the data and pushed it to the respective tables within the database.

5.3.1 Data Storage

To accommodate the analysis tasks described in Section 5.4, data generated throughout the execution and evaluation phase of the process depicted in Figure 5.2 are preprocessed and then stored in a relational database, which consists of two separate schemas. PostgreSQL[5] was used as RDBMS.

To make time series data accessible, it has to be linked with metadata that is recorded by researchers before the experiment and the quality data that is generated after the experiment. This data is stored in a schema (**meta data schema**, depicted in Figure 5.13) of the database. For each EXPERIMENT that was carried out, the respective identifiers of SENSORS that where deployed are stored. Among the attributes of each instance within the SENSOR table, datatable corresponds to the table name of the table within the time series schema that is used to store the data generated by the respective sensor. Deployments of sensors to experiments can change over time to make it possible to increase or decrease the number of sensors throughout the project.

Throughout each experiment, one or more BOREHOLES are drilled. BOREHOLES are assigned to exactly one experiment. Each BOREHOLE, however, can be drilled using different TOOLS and auxiliary systems such as SUCTION SYSTEMS or COOLING SYSTEMS. Furthermore, the drilling process associated with each borehole can potentially be separated into several phases. Each of these phases can have different process parameters, which are stored in the PHASES table. Every instance within the EXPERIMENTS, BOREHOLES and PHASES tables have attributes dedicated to storing their beginning and ending time points which can be used to find relevant periods within the tables storing time series generated by sensors.

Finally, QUALITY DATA for BOREHOLES are stored. The specimen used throughout the experiments can be composed of multiple layers, which reflects conditions typically found in the aviation industry. Each layer can have a thickness and material. A position attribute indicates the position of each layer within a given plate. This attribute is implemented as an integer, which increments for each layer, starting from the one closest to the tool. Respective data is stored in a dedicated table (LAYERS). It is important to note that quality measurements are not linked to a borehole but also the layers of each borehole. Therefore, a single borehole can have several measurement results concerned, for example, with its roundness assigned to it.

Time series data itself is stored in a separate schema (**time series schema**, depicted in Figure 5.12) is dedicated to storing only time series data in the form of simple timestamp-value pairs. Each

[4] http://www.ni.com/it-it/shop/compactrio.html
[5] https://www.postgresql.org/

sensor has a dedicated table assigned to it. The table name is used to identify the sensor that was used to generate the respective data. As can be seen, there are no relations between individual data tables in this schema. This schema is equivalent to a folder within a file system, where each sensor has a dedicated file to store its data. Only with the knowledge regarding which table corresponds to which sensor and at what time a given process happened the data can be accessed efficiently. Therefore, this schema on its own is not sufficient to make sensory data accessible for human users. It does, however, provide a scalable and flexible infrastructure to store potentially large amounts of data.

Figure 5.12 ERM of the time series schema.

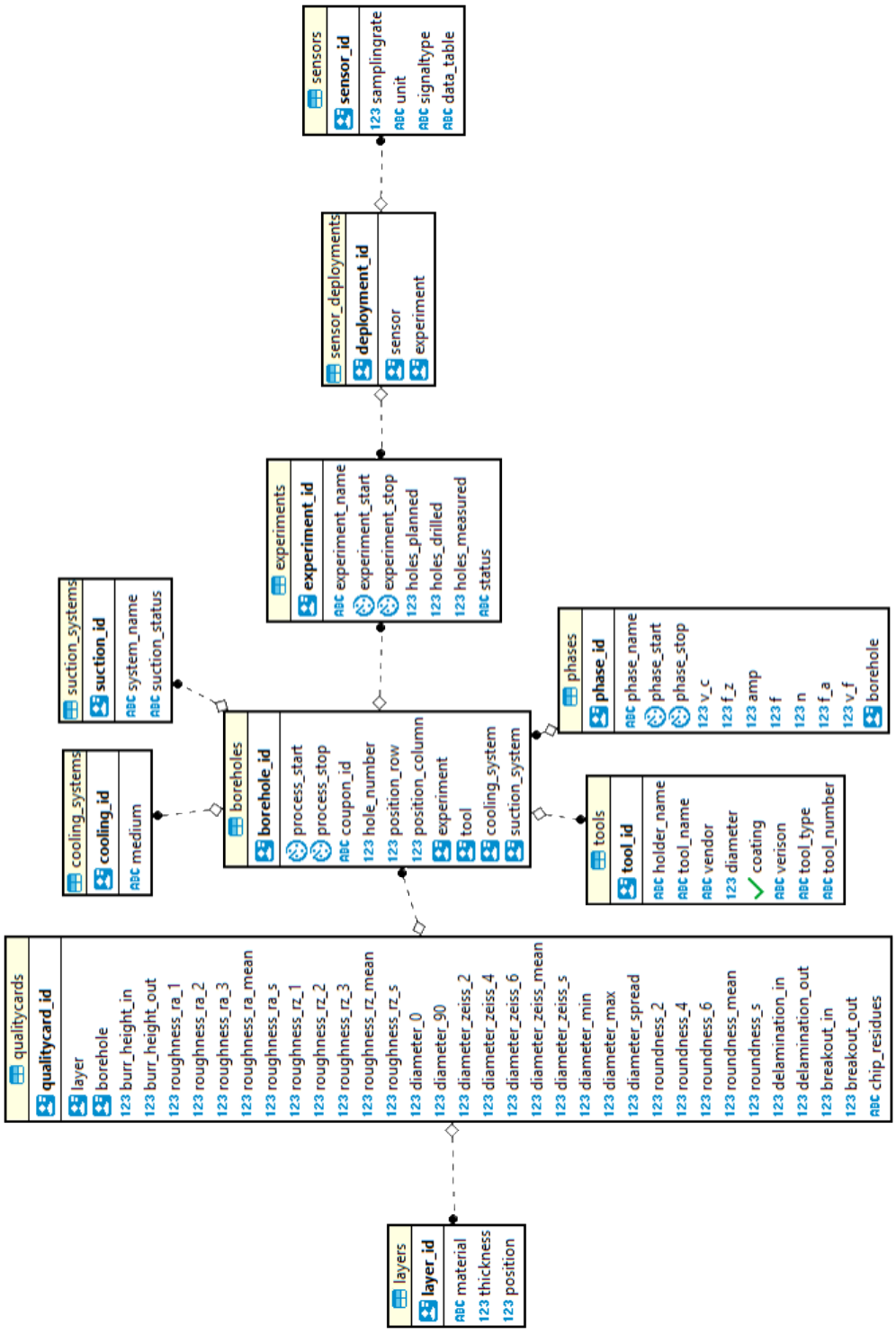


Figure 5.13 ERM of the meta data schema.

As of the writing of this thesis, the total size of the database is 1725 gigabyte. A vast majority of storage is occupied by time series data, and within that group, data from high-frequency sensors such as acceleration and sound measurements account for approximately two-thirds of the occupied storage space. In total, 1704 boreholes are stored in the database, 1200 of those have some quality measurements associated with them.

An interesting detail is the relatively low degree of utilisation of the storage space. Under the assumption that only those time series measurements that were created throughout manufacturing processes are of interest for analysis, based on the number of boreholes, the sampling rates and the average storage demand per sample, the theoretical minimum storage demand that would be needed can be calculated. Assuming an average process duration of 10 seconds, only approximately 13% of the actual database size would be required. The majority of the data, therefore, is not connected with manufacturing processes, but rather with downtimes. This illustrates how important it is to correctly select data and distinguish useful from non-useful data before analysing it.

5.3.2 Data Access

Generally, data stored in the database can be accessed through SQL queries. If, however, the data that should be queried is supposed to be time series data, data access becomes a two-stage process, which is illustrated in Figure 5.14. In the first step, the names of tables for relevant sensors as well as the temporal window within those tables are selected from the metadata schema based on user-defined filters. For example, a user might be interested in temperature measurements generated while titanium plates where machined. This would result in a table containing data table names of temperature sensors along with the (temporal) start and end points associated with the processes that are connected with titanium plates.

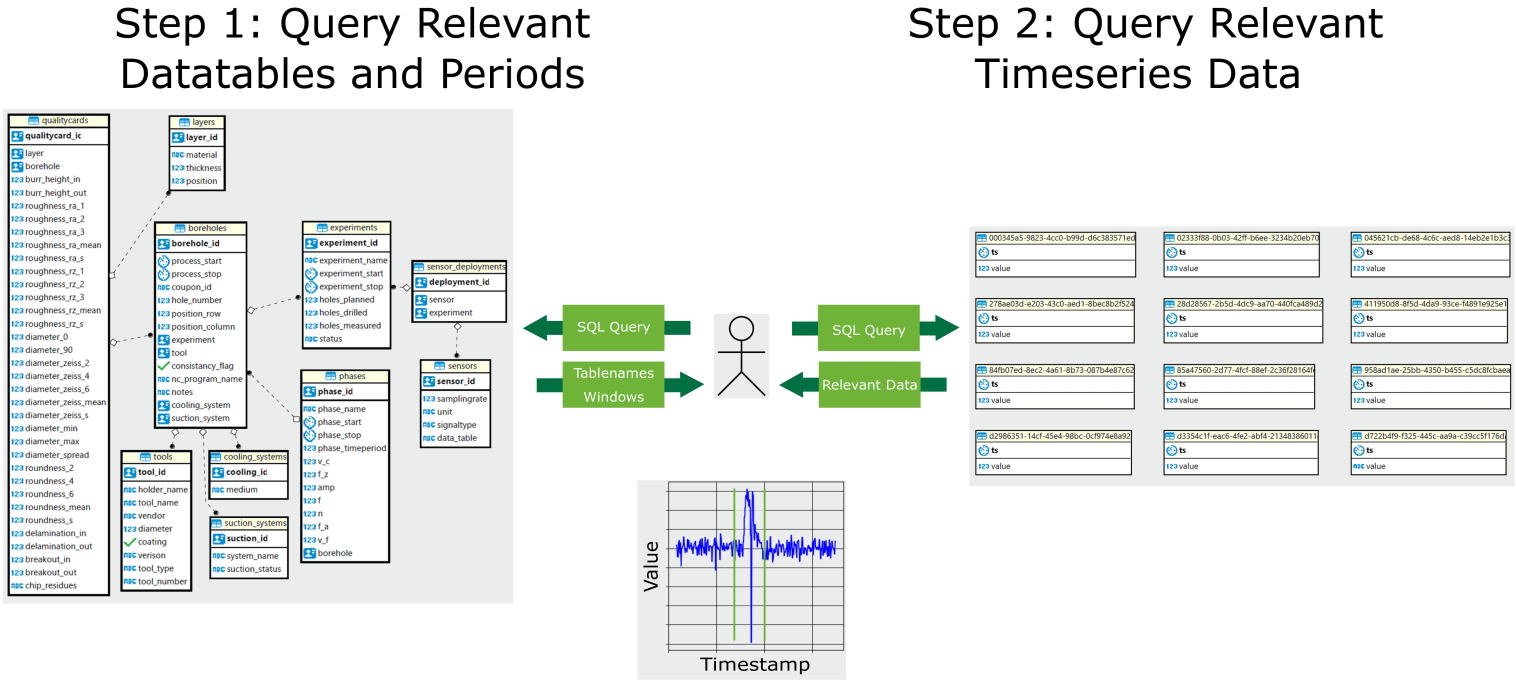


Figure 5.14 Approach used for data storage.

Table 5.2 Result of Query 5.1

data table	$process_{start}$	$process_{stop}$
b92b249e-2e...	2018-04-04 15:27:53	2018-04-04 15:28:14
ec941db6-ea...	2018-04-04 15:27:53	2018-04-04 15:28:14
b6cf547e-02...	2018-04-04 15:27:53	2018-04-04 15:28:14
80d20676-e0...	2018-04-04 15:27:53	2018-04-04 15:28:14
b92b249e-2e...	2018-04-04 15:22:36	2018-04-04 15:22:41
ec941db6-ea...	2018-04-04 15:22:36	2018-04-04 15:22:41

To illustrate this, consider for example, that for a particular data analysis task, force measurements are required. Only processes for a particular tool (MA-VII-V1-N11) and material (Titanium) combination should be retrieved. First, the respective process windows ($process_{start}$ and $process_{stop}$) as well as the respective data table names need to be retrieved. The query to do this is given in Listing 5.1.

```
01 | select distinct data_table, process_start, process_stop from meta_data
       .boreholes
02 | join meta_data.sensor_deployments on meta_data.boreholes.experiment=
       sensor_deployments.experiment
03 | join meta_data.qualitycards on qualitycards.borehole=borehole_id
04 | join meta_data.tools on tool=tools.tool_id
05 | join meta_data.layers on qualitycards.layer=layer_id
06 | join meta_data.sensors on sensor=sensor_id
07 | where (signaltype='ForceW1' or signaltype='ForceW2' or signaltype='
       ForceW3' or signaltype='ForceW4') and material='Titanium' and
       tool_name='MA-VII-V1-N11'
```

Listing 5.1 SQL query to retreive processes for given material (Titanium) and tool (MA-VII-V1-N11) combination

In the second step, for each of the processes found in the first step, the generated time series data has to be queried. Using the parameters "$process_{start}$", "$process_{stop}$" and "data table" from the previous query, the following query template can be used for this.

```
01 | select * from  time_series_data."data_table" where ts>'process_start'
       and ts < 'process_stop'
```

Listing 5.2 SQL query to retrieve time series data for a specific sensor

5.4 Data Analysis

Data such as the one presented here can be analysed in several different ways. In contrast to industrial applications, which typically have a relatively narrow set of questions to be answered, their potential

Table 5.3 Query Result for Query 5.3

	# of holes	$\bar{f}_{roundness}$	$\bar{f}_{roughness}$	$\bar{f}_{burrOut}$	$\bar{f}$
VAD	734	82.7	49.5	-8.8	41.1
no VAD	43	83.1	41.9	35.5	53.5

number is much higher in research applications (see Equation 1.1). In the following sections, some example data analysis tasks for both non-temporal and temporal data analysis will be showcased in order to define a baseline for the attempt to create a prototypical OBDA- based prototype, which will be described in Chapter 6.

5.4.1 Analysis of non-temporal Data

In the context of the application scenario, an instance of a non-temporal data analysis type is, for example, the investigation of the relation between quality criteria and process parameters.

A borehole can consist of several layers, each potentially made out of a different material. The set of layers per borehole is denoted as S_i. For each layer (j) of a borehole (i), quality measurements regarding specific features (n) can be determined ($x_n^{i,j}$). For each of these measurements, thresholds for the corresponding maximum allowed measurement values ($\hat{x}_n^j$) are defined in standards. For each material, different thresholds can be required.

Using these definitions, a measure for relative fitness of a particular borehole in respect to a given criterium (f_n^i) can be calculated. The overall fitness per feature per borehole is defined as the minimum fitness of all layer specific fitnesses of the respective feature (Equation 5.1a). Furthermore, an overall fitness value per borehole ($\bar{f}^i$) can be defined as the average value of all calculated, feature specific, fitness values (Equation 5.1b).

Based on this, all processes with axial oscillation amplitudes larger than zero could be defined as VAD processes. This was done in the query depicted in Listing 5.3. As can be seen in the result (Table 5.3), VAD processes seem to produce significantly worse quality results then conventional ones. This is against what would be expected from literature. This counter-intuitive result is due to the fact, that VAD processes where carried out mainly in experimental areas with unknown results, whereas conventional processes where carried out within well know process boundaries in order to create baselines.

$$f_n^{i,j} = \min_{\forall j \in S_i} \frac{\hat{x}_n^j - x_n}{\hat{x}_n} \tag{5.1a}$$

$$\bar{f}^i = \frac{\sum_n^k f_n^i}{k} \tag{5.1b}$$

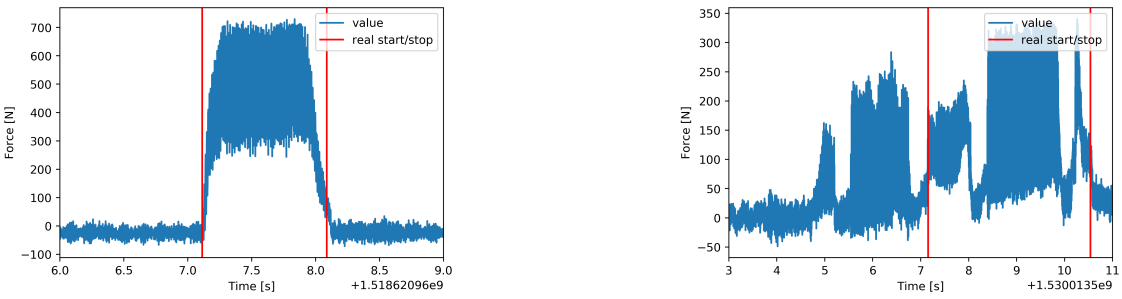
```
01 | select count(fitness_outer.borehole) as no_of_holes, avg(f_roundness)
       as f_roundness_avg,
02 | avg(f_roughness) as f_roughness_avg, avg(f_burr_out) as f_burr_out_avg
       , avg(f_all) as f_all_avg,
03 | case when (amp>0) then 1 else 0 end as VAD from (
04 | select borehole, min(f_roundness) as f_roundness, min(f_roughness) as
       f_roughness, min(f_burr_out) as f_burr_out,
05 | min((f_roundness+f_roughness+f_burr_out)/3) as f_all
06 | from (
07 | select borehole,
08 | case when roundness_mean is not null
09 | then (0.06-roundness_mean)/0.06*100
10 | else null end as f_roundness,
11 | case when roughness_ra_mean is not null
12 | then (1.6-roughness_ra_mean)/1.6*100
13 | else null end as f_roughness,
14 | case when burr_height_in is not null
15 | then case
16 | when material='Aluminum'
17 | then (0.127-burr_height_in)/0.127*100
18 | when material='Titanium'
19 | then (0.2032-burr_height_in)/0.2032*100
20 | else 0 end
21 | else null end as f_burr_in,
22 | case when burr_height_out is not null
23 | then case
24 | when material='Aluminum'
25 | then (0.127-burr_height_out)/0.127*100
26 | when material='Titanium'
27 | then(0.2032-burr_height_out)/0.2032*100
28 | else 0 end
29 | else null end as f_burr_out
30 | from meta_data.qualitycards
31 | join meta_data.layers on layers.layer_id=qualitycards.layer) as
       fitnesses_inner
32 | group by borehole) as fitness_outer
33 | join meta_data.phases on fitness_outer.borehole=phases.borehole
34 | where f_all is not null
35 | group by VAD
```

Listing 5.3 Query to calculate fitness per borehole for VAD processes and non-VAD processes

5.4.2 Analysis of Temporal Data

After the negative influence of VAD on the quality of manufactured boreholes was discovered, the question arose which other measures had a significant influence on quality. Apart from process parameters, there is the potential to investigate the relationship between time series data generated by different sensors and quality measures. Force measurements, for example, might be a good source of information ([67, 68, 139]). Therefore, for each manufacturing process, statistical aggregates (mean and standard deviation) should be calculated and correlated with one of the quality measures used in the previous section.

As can be seen in Figure 5.15, the process windows depicted by $\text{process}_{\text{start}}$ and $\text{process}_{\text{stop}}$ within the database are not exactly matching the start and end points of the actual drilling process. This is since these timestamps are generated automatically when the control program is executed on the machine. This program, however, also includes the approach of the tool to the workpiece. The section of the manufacturing process that determines the borehole quality, however, only starts when the tool first makes contact with the material. Consequently, before statistical aggregates can be calculated, real process windows $[s_i, t_i]$ need to be identified. This was done based on change-point detection. A change point is a sample or time instant at which some statistical property of a signal changes abruptly [115]. The property that was used here was the mean value. This rather inflexible approach can yield correct results for simple processes (Figure 5.15a), but also create false/inaccurate results in more complex cases (Figure 5.15b). Certainly, on a case to case basis, parameters can be changed to capture exceptions and improve results, but determining these (arbitrary) values is not sustainable and very time intensive.



(a) Single layer force measurement.

(b) Multi layer force measurement.

Figure 5.15 Force measurements with algorithmically determined process stars and stops.

Apart from aggregated force measurements, however, other sources can potentially be used to improve the performance of these models. For example, a predictor worth adding can be derived from electrical power measurements. Using Equations 5.2 the accumulated energy per tool (E_{tool}) can be calculated by summing up the integrals over all process power measurements generated for boreholes created with the same tool like the one under consideration. This measure is expected to

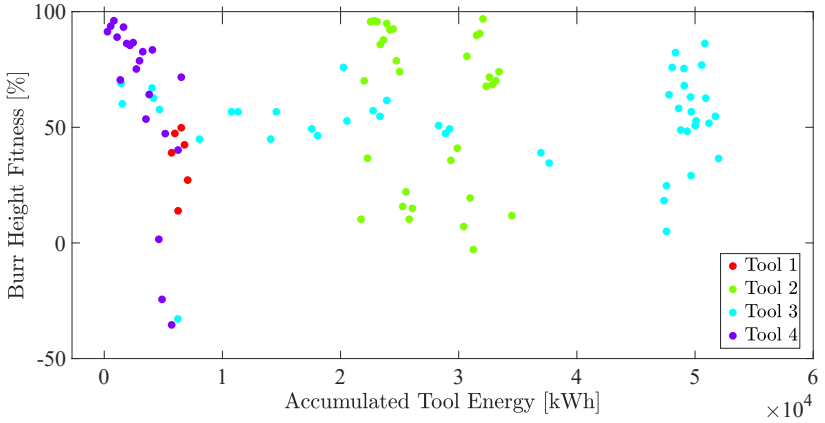


Figure 5.16 Fitness value vs. accumulated energy per tool for four example tools.

have a negative influence on the quality, as tool wear is expected to increase with the amount of energy converted by a particular tool. This relation, but also some anomalies can be seen in Figure 5.16.

$$E_{tool} = \sum_{i}^{k} \int_{s_i}^{t_i} P_{tool}(t)dt \tag{5.2a}$$

$$P_{tool} = P_{total} - P_{base} \tag{5.2b}$$

As the sample size (number of boreholes) in the data is relatively small (only 110 boreholes with sufficient quality data), a fairly simple linear regression model was chosen. Larger sample sizes would allow for the estimation of more sophisticated models and therefore would probably improve results. Nevertheless, the hypothesis that energy data actually contains useful information and therefore can be used to improve models holds, which is illustrated in Figure 5.18. Figure 5.17 depicts the fitting errors for tool 4. In comparison, the model which uses energy data achieves much smaller errors than the one which only relies on features acquired from force measurements.

5.5 Limitations of the Solution

The presented solution can, in general, be described as sufficient. Through the consistent application of state-of-the-art technology and defined processes, experimental data was collected, stored and analysed in a structured way. In the previous section, two types of data analysis and their results where shown. The first could be realised merely through SQL. For the second analysis, SQL had to be combined with some preprocessing in order to create necessary datasets. Specifically, 275 lines of MATLAB code where necessary to query data from the database, compute energies per process, accumulate process energies per borehole and clean the resulting datasets. Only 19 of those lines were

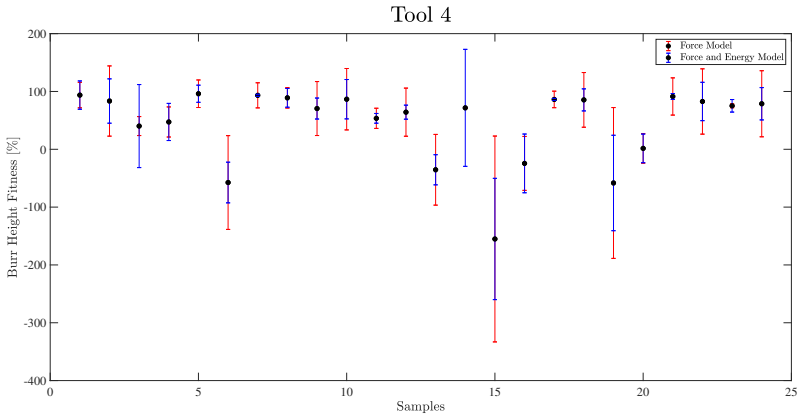
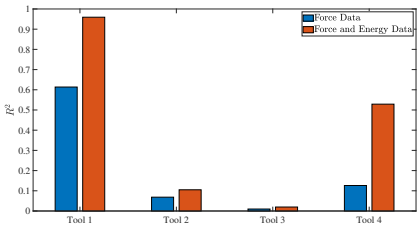
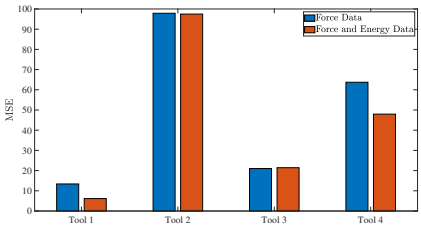


Figure 5.17 Fitness value vs. accumulated energy per tool for four example tools.



(a) R^2 for linear model predicting hole quality.



(b) Mean squared error (MSE) for linear model predicting hole quality.

Figure 5.18 Model quality metrics for different predictor sets.

needed for estimate model parameters. Not only is the amount of time which is required to write those lines comparably high, but also is the share between lines required for data access (256/ 93%) and analysis (19/ 7%) an indicator for how much of this work could potentially be reduced by improving the way data can be queried.

Apart from these discouraging numbers concerning efforts required to analyse data, also some other limitations exist for the current solution. In the following sections, these will be highlighted. Furthermore, for each of the identified limitations, potential alternative solutions will be described. As it turns out, already state of the art OBDA technology would significantly improve the existing solution. In order to accommodate all requirements, however, some additional features are needed, which will be included in the prototypical implementation in Chapter 6.

To give a better overview, the following requirements will be classified into three categories based on the necessary technologies to accommodate them. For each of the technologies, the respective Technology Readiness Level (TRL) is indicated. It can be observed, that this level decreases which means that the technologies necessary get less mature and therefore need an increasing amount of development work.

- OBDA (TRL 6-7)
- Temporal OBDA (TRL 4-5)

5.5.1 OBDA Level

Existing OBDA solutions can be used to accommodate some of the required features. The fact that this technology has matured significantly in the last few years opens the opportunity to solve some problems which appear (also) in the presented application scenario.

Data Integration and Data Reuse

In this application scenario, data was mainly added manually by researchers. To this end, for each experiment, relevant data describing the respective choice of tools, process parameters and materials were stored within tables which in turn were imported into the database. The same was done for quality measurements. Even though this was tedious labour, this approach worked well in the research environment. If applied in an industrial setting, not just one data source would have to be accessed, but several heterogeneous databases. Tool data, for example, might be stored in a dedicated tool management system. Data associated with more complex workpieces might come from CAD/CAM systems. According to Ekaputra et al. [61] and also Petersen et al. [142], OBDA is a feasible approach to integrate these data sources.

A variation of this challenge is the problem of inadequate data reuse. Data analysis tasks such as the ones described here can, however, be seen as detached from dedicated sets of experiments that lead to the generation of the data that they need if this data (or parts of it) already exists and is available for researchers. This would significantly reduce the work that has to be done in order to answer new research questions. This reuse of data seems obvious, especially as other scientific disciplines, such as medicine, computer science, experimental physics or political science have a much stronger history of applying quantitative methods then it is the case in manufacturing.

This shortcoming illustrates a structural problem that was already briefly mentioned at the beginning of this thesis. The way researchers tend to generate, store and access their data is highly individual, and interchanging datasets between different researchers are labour intensive in the best case and impossible in the worst case. As a consequence, experimental investigation currently is among the most popular research methods in the manufacturing domain even though its high costs. Currently, each research question leads to precisely one experiment and the data generated is used exclusively in the analysis that is connected with this particular question.

Imagine, for example, a research project in Japan that investigates the processing of different Titanium alloys. Through experiments, thousands of processes are carried out for different tools and process parameters. Similar to the presented application scenario, quality data and time series data is recorded. This data might potentially be interesting for researchers working within an industrial company that plans to use alloys that were used throughout these experiments. Optimal process

parameters could be identified and, more importantly, experiments could be designed in a way that prevents generating data that was already generated elsewhere. Currently, even if state-of-the-art database technologies were used, accessing the data would require other researchers first to understand the respective schema which would already be a labour intensive task without language barriers that can be expected in such a scenario. More importantly, the chance that researchers even find data that might be relevant for them is meagre. Indeed, literature research could lead to scientific publications from researchers who carried out relevant experiments, but realistically only a small fraction of data will be visible through this channel and the time delay between data generation and publication is significant. This problem is, at its core, another version of the data integration problem described at the beginning of this section. Experimental data generated and stored by different organisations are not compatible enough to access data in those systems collectively. Consequently, just as for industrial data integration tasks, OBDA could be used to make data generated by researchers accessible in a way that minimises necessary effort for researchers that were not involved in the original experiments. Through a shared, ontological layer, datasets and their respective meta-information could be made visible to other researchers more easily. Therefore, data could be accessed and reused more often, and the total number of experiments that need to be conducted could potentially be reduced significantly. OBDA seems to be a suitable way to overcome data access problems that lead to the limited reuse of experimental data in the manufacturing domain.

Knowledge Requirements

Data stored in relational databases, like the one that was used for the presented solution, can be accessed via SQL- queries. For many applications, these queries are predefined according to the respective application scenario and can be reused by users by simply exchanging some critical parameters of the query. This can be aided through user-friendly interfaces or other software. In this case, however, it can not be assumed that every possible query that a researcher might be interested in can be anticipated.

Consequently, researchers need to write SQL queries themselves when they need them. To be able to create such queries, however, knowledge regarding the database schema and, more importantly, the exact interpretation of the respective database table columns is required. Furthermore, some fundamental knowledge regarding the foundations of SQL syntax are necessary prerequisites. The most substantial part of the researchers who have to work with the database does not have any education in database theory which makes this requirement a significant issue.

This problem was the primary motivation for the development of OBDA technologies. OBDA allows users to access (potentially many) relational databases through queries which are formulated using terms and relations from their specific domain without having to care for the internal structure of the databases they are interacting with. These terms and relations are stored in the form of a domain-specific ontology. Consequently, with a suitable ontology at hand, OBDA would be a useful technology to improve interaction with data by domain experts.

5.5.2 Temporal OBDA Level

Some of the requirements that are motivated by this application scenario cannot be covered by conventional OBDA solutions. Only recently, temporal OBDA frameworks emerged, which allows us to apply them to the manufacturing domain.

Temporal Relations

Temporal aspects of the data we generate are important. Therefore, certain temporal facts such as times connected with processes or experiment starts and stops respectively are explicitly recorded in a certain table. These are considered as materialised facts. The selection of temporal intervals is a vital feature of the proposed solution. Criteria for those intervals, however, can only be relatively simple. For example, intervals are associated with processes. By filtering those processes, i.e. due to specific tool dimensions, workpiece materials of process parameter ranges, respective temporal intervals can be found.

In the research setting at hand, however, it can not be expected that this is a feasible approach for all potentially interesting time points. As soon as more complex selection criteria should be applied, the limitations of this system become apparent. For example, it is not possible to directly query intervals based on specific temporal orders of events. An example for such a definition might be "any intervals of force measurements which were recorded for drilling processes directly happening **after** aluminium was machined". Consequently, the ability to use temporal relations generated through reasoning to access data would be compelling.

Chapter 6

OBDA Prototype

In the following sections, a step towards implementations of (some) of the requirements formulated in the previous chapter will be taken in the form of a prototypical implementation.

This prototype is based on Ontop-temporal, a extension of Ontop [41], which was already described in Chapter 3. In alignment with the architecture of this tool, the prototype is separated into two parts. In the first part, a static ontology is used to model the respective domain of discourse and define all necessary static concepts. Then, extending this static ontology, some temporal rules will be defined. At the end of this chapter, an extension of the available languages is proposed based on the identified limitations of existing options.

6.1 Ontop-temporal

In principle, several OBDA systems are available for the implementation of this prototype. Only one of them, however, is both non-proprietary and, more importantly, offers temporal reasoning capabilities. As OBDA tool, Ontop exposes RDBMS as virtual RDF graphs. Ontop supports many relational database engines via JDBC. These include all major commercial relational databases (DB2, Oracle, and MS SQL Server) and the most popular open-source databases (PostgreSQL, MySQL, H2, and HSQL). Furthermore, Ontop can be used with federated databases to support multiple data sources (e.g., relational databases, XML, CSV, and Web Services). Ontop uses ontological knowledge to enrich queries and mappings to generate SQL queries automatically. As a consequence, SPARQL can be used to access the data stored in the linked databases.

Ontop-temporal allows the use of concepts formulated in $MTL^{nr}_{datalog}$ to capture temporal ontological knowledge. Such concepts, due to the chosen language, will be called rules. $MTL^{nr}_{datalog}$ was already introduced in Section 2.4.2.

6.1.1 Temporal OBDA Framework

Figure 6.1 depicts the OBDA framework which is at the basis of Ontop-temporal. Recall that in classical OBDA, a given OBDA specification is defined by a triple $\mathcal{P} = \langle \mathcal{O}, \mathcal{M}, \mathcal{S} \rangle$. In the case of temporal OBDA, this gets extended by temporal rules $\mathcal{T}$ as well as temporal mappings $\mathcal{M}_t$. This approach allows development of temporal and static ontologies independently from each other while still being able to combine them if necessary.

Just as in conventional OBDA, a user query is formulated in terms of concepts defined in the ontology layer. The connection between the ontology layer and the data sources is still established through mappings. Now, however, these mappings also include temporal mappings which are used to assign validity periods to data instances.

Concrete examples on the basis of the introduced application scenario illustrating these two parts will be presented in Sections 6.2 and 6.3, respectively. Before, however, the system architecture of Ontop-temporal will be described.

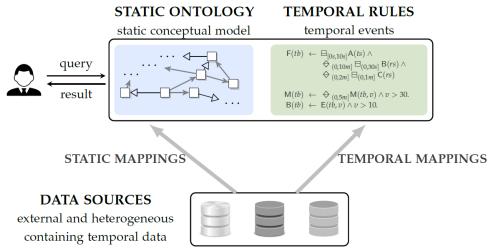


Figure 6.1 Temporal OBDA Framework as defined by Kalaycı [92].

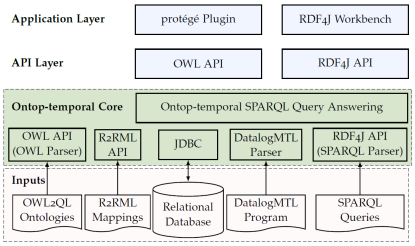


Figure 6.2 System Architecture of Ontop-temporal [92].

6.1.2 Ontop-temporal Architecture

Figure 6.2 depicts the system architecture of Ontop-temporal. It consists of the following four layers [92]:

1. the inputs, which are the domain-specific artefacts such as the ontology, $MTL^{nr}_{datalog}$ program, database, mappings, and queries;
2. the core of the system in charge of query translation, optimisation, and execution;
3. the Application Programming Interface (API)s exposing interfaces to users of the system; and
4. the applications that allow end-users to execute SPARQL queries over databases.

Standard W3C recommendations related to OBDA (such as OWL2QL, R2RML, SPARQL, and the OWL2QL entailment regime in SPARQL) are supported and a Protégé plugin is available. This plugin will also be used for our prototype.

In addition, the use of non-recursive Datalog rules is allowed. The syntax for $MTL^{nr}_{datalog}$ introduced the operators $\boxplus$, $\boxminus$, $\diamondplus$, $\diamondminus$, $\mathcal{S}$ and $\mathcal{U}$ as ALWAYS IN FUTURE, ALWAYS IN PAST, SOMETIME IN FUTURE, SOMETIME IN PAST, SINCE, and UNTIL.

R2RML is used as a mapping language. Mappings create named graphs to represent temporal information. Alternatively, it is also possible to use the Ontop mapping language to provide a more compact syntax.

6.2 Static OBDA

As was described in Section 5.4, not all data analysis tasks in the manufacturing domain require handling of time series data. For those kinds of analysis, conventional (non-temporal) OBDA solutions suffice. In this section, such a solution is presented. The main part of this section is the development of a domain specific ontology. Based on this ontology, a working OBDA prototype is implemented to solve the following requirements, as identified in the previous chapter, was implemented:

- Data integration and reuse
- Knowledge requirements

The capability of OBDA to integrate different physical data sources within a single organisation using a single ontology was already shown by Petersen et al. [142]. Another aspect of this, however, was not illustrated by the authors. In order to integrate data across organisational borders, which is a prerequisite for most data reuse scenarios, different perspectives need to be taken into account. The same piece of data might be referred to as something semantically different based on the respective users' background. As described by Ekaputra et al. [61], combinations of different ontologies can be used to accommodate this need. In the following section, this approach will be illustrated by showing how different ontologies, each with an increasing degree of specialisation, can be used for the presented application scenario.

Static Ontology

To be able to use OBDA principles in the presented context, first an ontology needs to be developed. Two levels of detail were covered throughout the ontology development process. On the first level, the goal was to create a common vocabulary for the manufacturing domain which is relatively independent of project-specific terms. This proposal for a unifying view on processes within the manufacturing domain is closely related to another, highly popular ontology: Semantic Sensor Network Ontology (SSN). This general vocabulary can then be used to import lower level ontologies (such as the one developed for the presented use case) and therefore enable to use terms and concepts that are project specific while still providing interoperability between different projects through the top-level ontology. In the following sections, the ontologies will be described in more detail.

Semantic Sensor Network Ontology

SSN is an ontology which was developed to describe observations made by sensors in a general (domain independent) way. It is divided into two parts, SSN which represents a detailed view and Sensor, Observation, Sample, and Actuator (SOSA), which is a subset, containing only the most commonly used concepts. The goal of this ontology is to facilitate the interaction of data generating devices from a wide range of application areas. The respective concepts and their relations are depicted in Figure 6.3.

Not all of the depicted concepts are directly relevant for the scope of this thesis. Those that are however, are SENSORS, OBSERVATIONS, FEATURESOFINTEREST and PROPERTIES. Furthermore, the concepts SYSTEM and PLATFORM will be reused.

In the context of SSN, FEATURESOFINTEREST can be anything physical (a tree, a car, a group of people) which is of interest for some application. FEATURES have PROPERTIES, which can be observed to directly or indirectly (through calculation) arrive at results. For example, a property of a car might be its fuel consumption.

OBSERVATIONS are generated by SENSORS and are associated with PROPERTIES of FEATURESOFINTEREST. For example, a given SENSOR can repeatedly measure the temperature within a room. Each measurement would in this case be an OBSERVATION of the Property "Temperature" of the FEATURESOFINTEREST "Room". The result of such an OBSERVATION would be a single numeric value. Such Results, however, can also be more complicated. Other forms of results might be images or structured data.

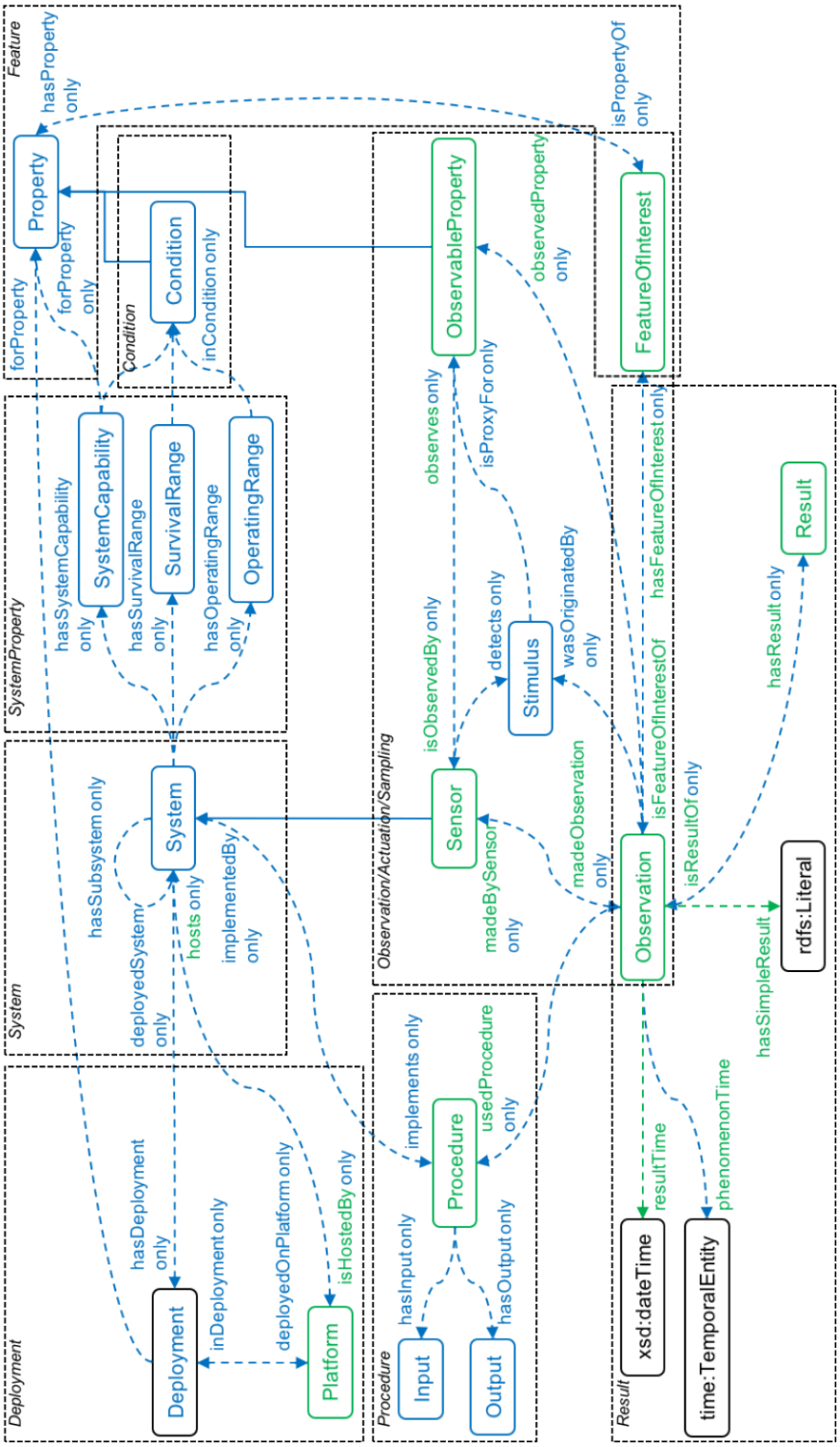


Figure 6.3 Depiction of concepts and relations of the SSN ontology concerned with observations of sensors.

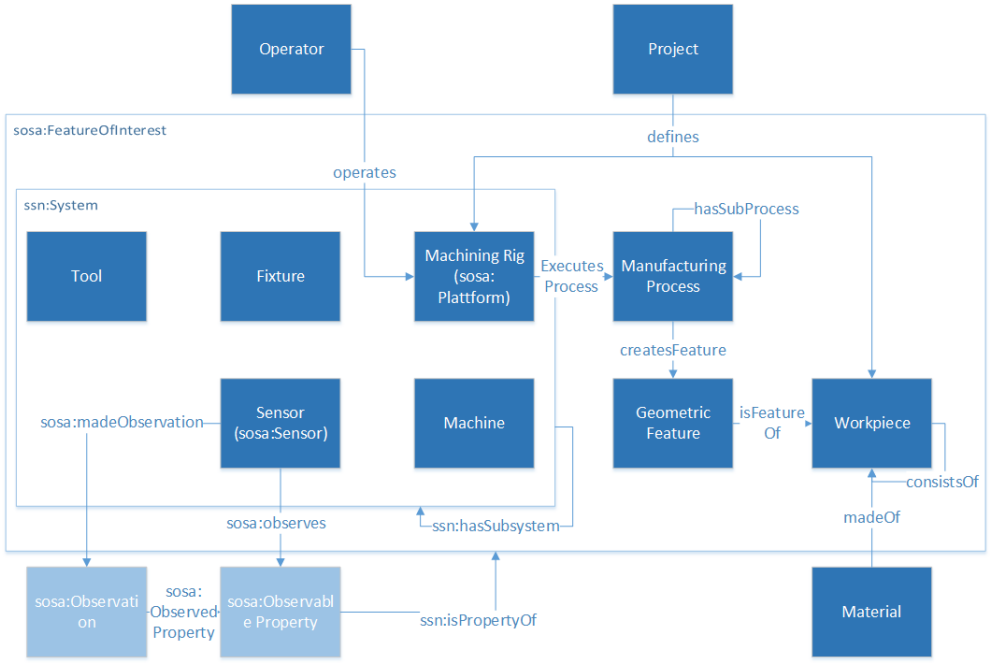


Figure 6.4 Top level Production Systems (ps) ontology to describe general concepts in the domain of manufacturing processes.

Production Systems Ontology

Based on the previously described SSN ontology, another ontology with domain-specific terms for the manufacturing domain was developed. The scope of this ontology was, to be able to talk and reason about concepts that are concerned with manufacturing processes and their outcomes. In the form of MASON [117], a similar, but slightly different ontology exists. The difference is the emphasis on the process nature and the necessity to associate sensor readings (observations) with those processes as well as their outcomes.

The result of the development process is depicted in Figure 6.4. In the centre of the ontology stand MANUFACTURING PROCESSES. Such processes can, following DIN 8550 be further distinguished into one of the six main process groups Change Properties, Coating, Joining, Cutting, Forming and Primary Forming [73]. Furthermore, processes can themselves have subprocesses which makes it possible to describe more sophisticated manufacturing processes which often cover several different process steps. Potentially, this can also be used to distinguish phases within manufacturing processes. In the case of drilling, one might, for example, want to describe that a given drilling process is the combination of two process steps (tapping before drilling). The potential to use this to describe temporal dependencies of processes is obvious and will be further discussed below.

The result of any MANUFACTURING PROCESS is a GEOMETRIC FEATURE, which is associated with a particular WORKPIECE. GEOMETRIC FEATURES are particular kinds of shapes that workpieces might have such as holes, corners and pockets.

MANUFACTURING PROCESSES are executed by MACHINING RIGS. Those are PLATFORMS in the sense of the SSN ontology and are (potentially) complex combination of SYSTEMS such as PRODUCTION MACHINES, FIXTURES, TOOLS. Some of these systems can also be considered to be SENSORS (i.e. they generate data).

Each MACHINING RIG has at least one assigned OPERATOR, which is a natural person in charge of carrying out processes on the specific rig. Furthermore, the concept of PROJECT is used to describe entities that motivate the manufacturing of a set of workpieces and potentially define the required MACHINING RIGS.

Finally, as can be seen, almost all of the defined concepts are also considered to be FEATURES OF INTEREST, which makes it possible to define PROPERTIES which might be subject to OBSERVATIONS. For example, a WORKPIECE "Test Specimen" might have several GEOMETRIC FEATURES "Boreholes", which in turn have PROPERTIES such as "Surface Roughness" or "Roundness" which might be observed through an OBSERVATION "Quality Measurement".

This ontology can also be described more concisely using DL. The respective assertions not covered by SOSA or SSN can be seen in Equation 6.1.

$$
\begin{aligned}
Tool &\sqsubseteq System \\
\exists operates &\sqsubseteq Operator \\
Fixture &\sqsubseteq System \\
Sensor &\sqsubseteq System \\
MachiningRig &\sqsubseteq System \\
\exists operates^{-} \sqcup \exists defines^{-} \sqcup & \\
\exists executesProcess &\sqsubseteq MachiningRig, Machine \sqsubseteq System \\
\exists defines &\sqsubseteq Project \\
\exists executesProcess^{-} \sqcup \exists hasSubProcess^{-} \sqcup & \\
\exists hasSubProcess \sqcup \exists createsFeature &\sqsubseteq ManufacturingProcess \\
\exists createsFeature^{-} \sqcup \exists isFeatureOf &\sqsubseteq GeometricFeature \\
\exists isFeatureOf^{-} \sqcup \exists defines^{-} &\sqsubseteq Workpiece
\end{aligned}
$$

Clean Drilling Ontology

Until now, only general concepts and relations from the manufacturing domain were considered. In order to be able to cover project-specific terminology, however, more specialised ontologies need to be developed. Those can then be combined with the top level production systems ontology and, in turn, make data exchange over project boundaries possible. An example of such a specific ontology was developed based on the requirements of the application scenario described above (Section 5).

This ontology will, according to the name of the motivating project, be called clean drilling ontology. The resulting ontology reuses and expands concepts of the production systems ontology.

Each of the depicted concepts are subclasses of more general concepts within the production systems ontology. Apart from trivial expansions such as BOREHOLE as a specific kind of GEOMETRIC FEATURE or Plate as a particular kind of WORKPIECE, this ontology introduces a special kind of MANUFACTURING PROCESS. Specifically, VAD is defined as any drilling process where the set vibration amplitude was above zero. This rule can be implemented through respective mappings (as will be described below).

Furthermore more subclasses of the OBSERVABLE PROPERTY class are introduced. QUALITY PROPERTIES are all properties which are connected with the quality of GEOMETRIC FEATURES, as described in the example above. PROCESS PROPERTIES, on the other hand, are properties of MANUFACTURING PROCESSES. These properties might be working area temperature, electrical power input or workpiece forces which are measured by dedicated sensors.

The mentioned concepts can, similar to what has been done for those in the more general production systems ontology, be expressed in DL. The respective terms given here are only a small part of the full ontology, which can be found in Appendix A.

$$
\begin{aligned}
Drill &\sqsubseteq Tool \\
CoatedDrill &\sqsubseteq Drill \\
CuttingProcess &\sqsubseteq ManufacturingProcess \\
DrillingProcess &\sqsubseteq CuttingProcess \\
VADProcess &\sqsubseteq DrillingProcess \\
Borehole &\sqsubseteq GeometricFeature \\
Plate &\sqsubseteq Workpiece \\
QualityProperty &\sqsubseteq ObservableProperty \\
ProcessProperty &\sqsubseteq ObservableProperty
\end{aligned}
$$

Ontology Implementation

In order to be compatible with the Ontop platform, the aforementioned ontologies had to be implemented using OWL. This was done using the Protégé- plugin. In Listing 6.1, some example declarations, capturing core concepts, are listed. The prefixes PS and CD are used as shortcuts for the ontology Internationalized Resource Identifier (IRI)s <http://ontologies.ift.at/production_systems.ttl/0.5/> and <http://ontologies.ift.at/clean_drilling_ontology/0.1/> respectively.

In this example, the concept PS:MACHINE is defined as a subclass of the more general concept from the SSN ontology SSN:SYSTEM. Furthermore, the property PS:EXECUTESPROCESS is defined. Here, also domain and range are given as PS:MACHININGRIG and PS:MANUFACTURINGPROCESS

```
ps:Machine rdf:type owl:Class ;
rdfs:subClassOf ssn:System .
ps:executesProcess rdf:type owl:ObjectProperty ;
rdfs:domain ps:MachiningRig ;
rdfs:range ps:ManufacturingProcess .
cd:Borehole a owl:Class ;
rdfs:subClassOf ps:GeometricFeature .
cd:Plate a owl:Class ;
rdfs:subClassOf ps:Workpiece .
```

Listing 6.1 Example OWL axioms

```
mappingId urn:VADProcesses
target :process/{borehole_id} a cd:VibrationAssitedDrilling .
source select distinct borehole_id from meta_data.boreholes
join meta_data.phases on borehole_id=phases.borehole
where amp>0
```

Listing 6.2 Mapping between VAD Process concept and the respective rows in the boreholes table

respectively. The last two assertions link concepts from the project specific clean drilling ontology to the more general production systems ontology. CD:BOREHOLE is defined as a subclass of PS:GEOMETRICFEATURE and CD:PLATES as specific PS:WORKPIECES.

Mappings

Mappings are used to connect the previously described ontologies with the data stored in the database. An example mapping, which is used to define the concept VIBRATIONASSISTEDDRILLINGPROCESS is shown in Listing 6.2. The source definition is used to filter out only those PROCESSES, which had any process phase with a pecking amplitude (amp) larger then zero.

A more complex mapping has to be used for the connection between general BOREHOLES and their respective fitness values, as they were defined in the previous section. The implementation can be seen in Listing 6.3.

The examples show that already the definition of mappings can be used to enrich queries with explicit domain knowledge such as "any process with at least one phase with a pecking amplitude larger then zero is a vibration assisted drilling process", or the material specific threshold in the second mapping. This takes away the task of finding the exact numerical values for such classifications from the user and stores them in a single place. If at some point in the future these definitions would change, they can easily be adapted. The users do not necessarily need to be aware of this and can still us their well know vocabulary to access data.

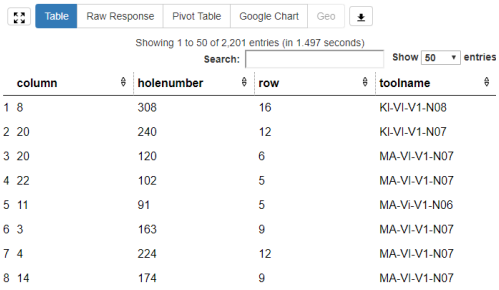
```
01 | mappingId urn:Boreholes
02 | target :borehole/{borehole_id} :isFeatureOf :workpiece/{coupon_id} ; cd:
       holenumber {hole_number} ; cd:positionRow {position_row} ; cd:
       positionColumn {position_column} ;ssn:hasProperty :diameter/{borehole_id}
        , :roughness/{borehole_id} , :roundness/{borehole_id} , :burrheight/{
       borehole_id} , :delamination/{borehole_id} ,:breakout/{borehole_id} , :
       chipresidues/{borehole_id} ; cd:roundnessFitness :fitness/{f_roundness};
       cd:roughnessFitness :fintess/{f_roughness}; cd:burrHeightOutFitness :
       fitness/{f_burr_out}; cd:overallFitness :fitness/{f_all} .
03 | source select distinct borehole_id, coupon_id, hole_number, position_row,
       position_column, f_roundness,f_roughness, f_burr_out, f_all
04 | from (
05 | select fitnesses_inner.borehole, min(f_roundness) as f_roundness, min(
       f_roughness) as f_roughness, min(f_burr_out) as f_burr_out,
06 | min((f_roundness+f_roughness+f_burr_out)/3) as f_all
07 | from (
08 | select qualitycards.borehole,
09 | case when roundness_mean is not null
10 | then (0.06-roundness_mean)/0.06*100
11 | else null end as f_roundness,
12 | case when roughness_ra_mean is not null
13 | then (1.6-roughness_ra_mean)/1.6*100
14 | else null end as f_roughness,
15 | case when burr_height_in is not null
16 | then case
17 | when material='Aluminum'
18 | then (0.127-burr_height_in)/0.127*100
19 | when material='Titanium'
20 | then (0.2032-burr_height_in)/0.2032*100
21 | else 0 end
22 | else null end as f_burr_in,
23 | case when burr_height_out is not null
24 | then case
25 | when material='Aluminum'
26 | then (0.127-burr_height_out)/0.127*100
27 | when material='Titanium'
28 | then(0.2032-burr_height_out)/0.2032*100
29 | else 0 end
30 | else null end as f_burr_out
31 | from meta_data.qualitycards
32 | join meta_data.layers on layers.layer_id=qualitycards.layer) as
       fitnesses_inner
33 | group by borehole) as fitnesses_outer
34 | join meta_data.qualitycards on fitnesses_outer.borehole=qualitycards.borehole
35 | join meta_data.layers on layers.layer_id=qualitycards.layer
36 | join meta_data.boreholes on fitnesses_outer.borehole=boreholes.borehole_id
```

Listing 6.3 Mapping connecting general Boreholes concept with respective database rows without the use of a view

SPARQL Endpoint

Even though Ontop also provides a Protégé plugin that can be used to query data through SPARQL queries, the prototype was implemented differently. Using a docker container which is provided by the developers of Ontop[1], a web-based interface to the database was created. Through this, SPARQL queries can be created in an interactive editor (Figure 6.5a). Those queries can then be executed over the database. The result is displayed in the tool as well (Figure 6.5b). The example query depicted in the aforementioned figures queries boreholes that were created using only a specific kind of tools.

(a) SPARQL query editor.



	column	holenumber	row	toolname
1	8	308	16	KI-VI-V1-N08
2	20	240	12	KI-VI-V1-N07
3	20	120	6	MA-VI-V1-N07
4	22	102	5	MA-VI-V1-N07
5	11	91	5	MA-Vi-V1-N06
6	3	163	9	MA-VI-V1-N07
7	4	224	12	MA-VI-V1-N07
8	14	174	9	MA-VI-V1-N07

(b) Results for the example query as displayed by the endpoint.

Figure 6.5 Screenshot from the SPARQL endpoint.

6.3 Temporal OBDA

Just as for static concepts, in this section temporal rules capturing ontological knowledge will be presented. Through temporal mappings, these rules can be linked to the data source. As mentioned, a Protégé plugin aiding this process is available. In the following sections, snippets from the implementation process will be shown. The full rules and mappings can be found in Appendix A. Not all necessary concepts can be modelled using the language currently provided by Ontop-temporal. Comments on necessary extension will be given in Section 6.5.

6.3.1 Temporal Rules

Before actual temporal rules are defined, some non-temporal rules are needed. Specifically, a VAD-phase is defined using the set axial pecking frequency. These values are recorded in a dedicated table in the database along with other process parameters. It has to be noted that these values are not measured values. Even though such a concept was already defined above, in order to make the temporal rules less verbose, an equivalent definition is given here in the form of a rule.

[1]The image is available here: https://hub.docker.com/r/ontop/ontop-endpoint

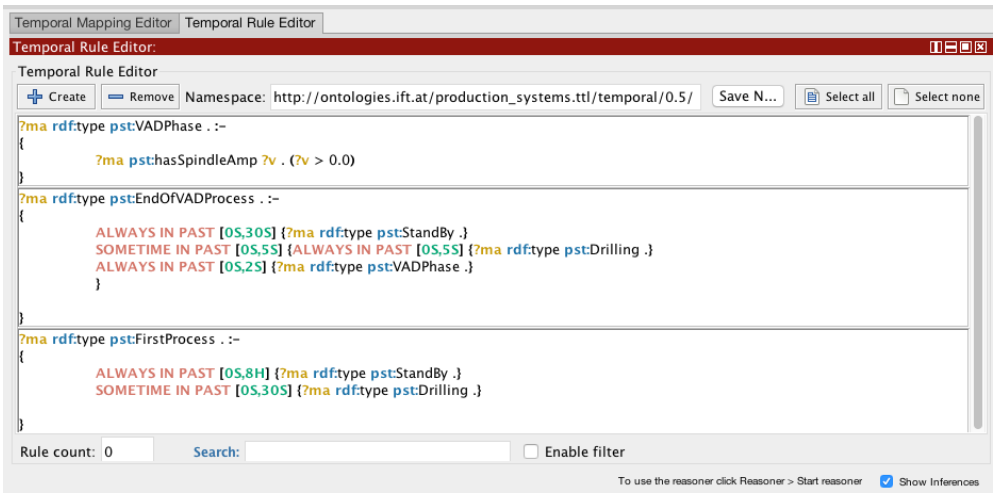


Figure 6.6 Screenshot from Protégé temporal rule editor.

$$\text{VADPhase}(ma) \leftarrow \text{spindleAmplitude}(ma, Amp), Amp > 0.$$

Using the rule given above, temporal concepts can be defined. The FIRSTPROCESS after a long STANDBY period (mostly this will be the first process of a day) can, for example, be defined using the operators $\boxminus$ and $\diamondminus$. The respective rule (6.3a) can be translated into natural language as "the first process on a machine is any period that is a drilling process which lasts at least 30s after a standby period that lasted at least 8h".

Similarly, a slightly more complicated concept, ENDOFVADPROCESS can be defined using another rule. This concept could be used to find periods where a longer standby period follows a production process that was a VAD process (6.3b).

$$FirstProcess(ma) \leftarrow \boxminus_{[0,8h]} Standby(ma), \diamondminus_{[0,30s]} Drilling(ma). \tag{6.3a}$$

$$EndOfVADProcess(ma) \leftarrow \boxminus_{[0,30s]} Standby(ma), \diamondminus_{[0,5s]} [\boxminus_{[0,5s]} Drilling(ma), \boxminus_{[0,2s]} VADPhase(ma)]. \tag{6.3b}$$

Figure 6.6 shows the implementation of some of these rules using the rule editor, which is available as Protégé plugin.

6.3.2 Temporal Mappings

Figure 6.7 shows a screenshot from the temporal mapping editor which is, again, available in the form of a Protégé plugin. As can be seen, the database that was used as a data source for those mappings has a slightly different structure as the one described in Section 5.3. This is because the original

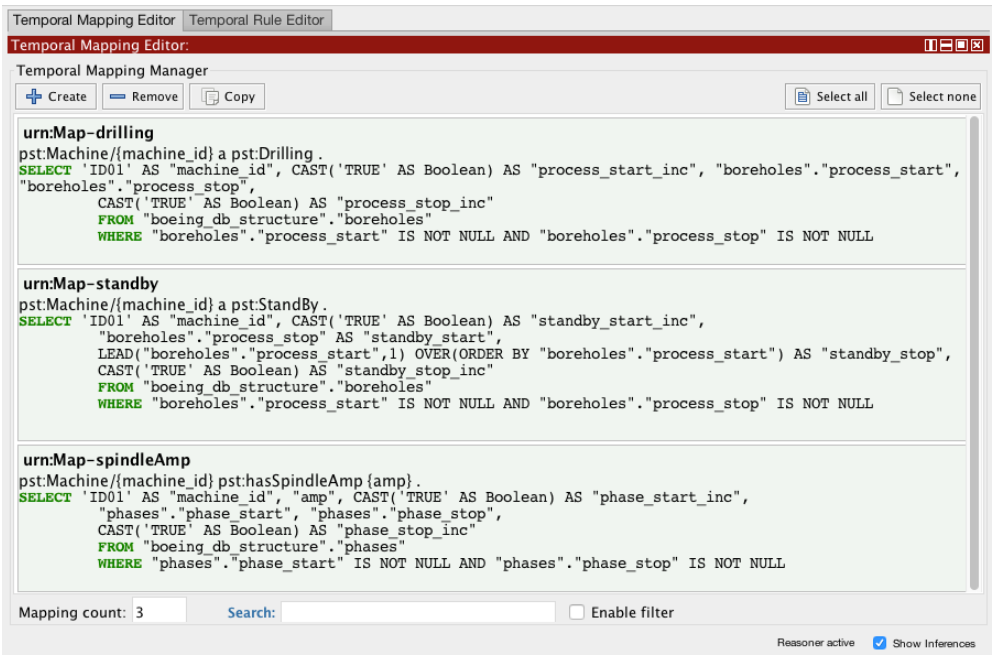


Figure 6.7 Screenshot from Protégé Mapping Editor Dialog.

database ran a version of PostgreSQL which does not support SQL commands necessary for the mappings.

Furthermore, notice, that two concepts DRILLING and STANDBY are defined through temporal mappings. As drilling periods are recorded explicitly through the attributes $\text{process}_{\text{start}}$ and $\text{process}_{\text{stop}}$ respectively, this can be done directly by assigning those values to be discriminators for the validity intervals. In the case of STANDBY, the periods between two subsequent processes were defined as validity intervals.

6.4 Prototype Evaluation

Using the example data analysis tasks from Section 5.4, the practical usability of the proposed approach can be evaluated.

6.4.1 Non Temporal Evaluation

As the first evaluation scenario, the basic data access use case presented in Section 5.3.2 will be used. Remember, that in the original system, data access was a two-stage process. In the first stage, a temporal window within a sensor data table was selected based on metadata connected with the drilling process. Data within this window was then queried.

In practice, in order to give domain experts the ability to interact with the database, their natural language queries have to be translated to respective SQL queries. Consider, for example, the following query (Listing 6.4), which can be used to replace the first step of the data access process described

```
01 | PREFIX ps: <http://ontologies.ift.at/production_systems.ttl/0.5/>
02 | PREFIX cd: <http://ontologies.ift.at/clean_drilling_ontology/0.1/>
03 | PREFIX ssn: <http://www.w3.org/ns/ssn/>
04 | PREFIX sosa: <http://www.w3.org/ns/sosa/>
05 | PREFIX time: <http://www.w3.org/2006/time#>
06 | SELECT ?start ?end ?datatable
07 | WHERE
08 | {
09 | ?rig ps:executesProcess ?process;
10 | sosa:hosts ?sensor.
11 | ?sensor a cd:Microphone;
12 | cd:datatable ?datatable.
13 | ?process a cd:VibrationAssitedDrillingProcess;
14 | time:hasBeginning ?start;
15 | time:hasEnd ?end.
16 | }
```

Listing 6.4 Evaluation Query

in Section 5.3.2. As can be seen, it is not necessary to know anything about the database schema. Furthermore, SPARQL is arguably a more intuitive query language then SQL. Using the developed OBDA prototype, however, it seems much more realistic that domain experts could be able to query data themselves without having first to consult database developers.

Another evaluation scenario is the first data analysis task (see Section 5.4.1). Fitness values for each borehole where evaluated. The goal was to check whether or not VAD processes generate significantly better results then non-VAD processes. This analysis was done using a single SQL query. With 35 lines, however, this query was relatively complex, and it can be considered unlikely that manufacturing domain experts would be willing to write such a query.

An example query, which is used to access all fitness values for VAD-processes is shown in Listing 6.5. Using OBDA, however, the aforementioned static ontology can be used to access data without having to create such a query. The necessary SQL queries to access the data are hidden in the mappings. This makes it possible to directly query data without having to know about the underlying database schema or write lengthy SQL queries. Furthermore, the exact criterion that is used to distinguish VAD processes from non-VAD processes ($amp = 0$) does not need to be known to the user. Similarly, the material dependent thresholds that were used for calculating fitness values are not needed for querying. The result of this query, however, is not aggregated yet. Even though SPARQL provides this feature, it is not supported by Ontop yet. Consequently, this step of the data analysis task needs to be done using an external tool. Furthermore, a similar query to the one listed would have to be made for non-VAD processes respectively.

For both examples, querying became simpler than in a conventional system. Nevertheless, text-based query formulation through a SPARQL endpoint is still a considerable barrier for domain experts.

```
PREFIX ps: <http://ontologies.ift.at/production_systems.ttl/0.5/>
PREFIX cd: <http://ontologies.ift.at/clean_drilling_ontology/0.1/>
SELECT ?overallFitness ?roughnessFitness ?burrheightFitness
?roundnessFitness
WHERE
{
?process a cd:VibrationAssitedDrillingProcess;
ps:createsFeature ?borehole .
?borehole cd:overallFitness ?overallFitness;
cd:roughnessFitness ?roughnessFitness;
cd:burrHeightOutFitness ?burrheightFitness;
cd:roundnessFitness ?roundnessFitness.
}
```

Listing 6.5 Evaluation Query

As was shown within the Optique project, visual, web-based, query tools can, however, be developed in order to compensate this.

6.4.2 Temporal Evaluation

Using the FIRSTPROCESS concept, the following query can be used to single out hole fitnesses in order to see whether or not they deviate significantly from those achieved on average. Potentially, this could happen due to changing temperature condition throughout the first few processes before the machine eventually reaches its steady state. The respective query is shown in Figure 6.8.

The result of this query, which is depicted in the same figure was generated at a reasonable time (8s). A comparison with the equivalent SQL query, which is automatically generated by Ontop-temporal (Appendix B) shows, that temporal OBDA can potentially dramatically reduce the time needed to access data. The result itself, however, only contains repetitions of the same process and therefore is not meaningful. It was not possible to reconcile this problem quickly.

More importantly, using the available constructs, the analysis task described in Section 5.4.2 could not be implemented using Ontop-temporal which is because the expressivity of the available rule language is not high enough. Consequently, instead of trying to improve the performance of the temporal OBDA prototype, work on an alternative, more expressive language was started by the author. Based on domain requirements, defined in the form of use cases, a formal language was developed. This formal language was then used as the basis for an implementation of a prototypical system which was evaluated using data from the aforementioned database. The result of this is presented in the following section.

6.5 Accessing Time Series Logs

After analysing existing solutions to accessing temporal data ontologically, it became apparent that several important concepts cannot be expressed using the available languages. In particular, this is

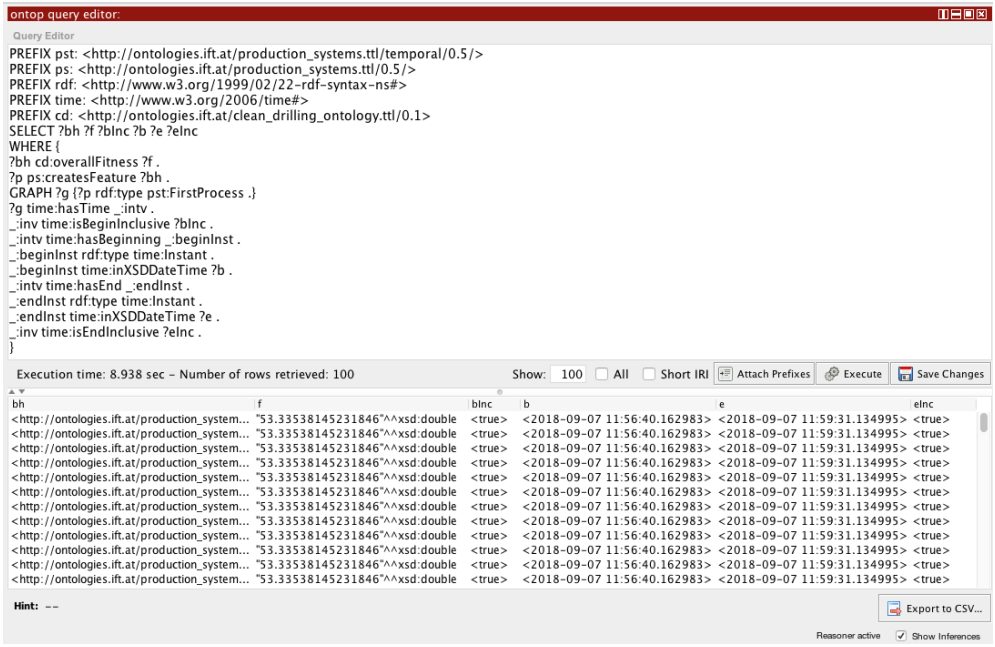


Figure 6.8 Temporal query execution.

because current solutions are not suitable for direct access to time series data. To refer back to the terminology introduced in Section 2.1, current solutions can improve access to event logs but are not compatible with time series logs.

In order to further investigate this, it is useful to discuss the way we look at time series data. For this, consider the Figure 6.9. Depicted here is what would be the expected course of tool force measurements throughout a drilling process, which can be defined as a sequence of three sub-processes following each other. First, the drill makes contact with the workpiece, and the borehole is created (TAPPING). In this phase, the diameter increases, which results in increasing process forces. Then, the diameter, and therefore the process forces stay almost constant (DRILLING) until the drill breaks through on the other side of the plate, which results in a decrease of process forces (EXITING).

This description can be seen as the result of a "mental simulation" of the expected process, carried out by a domain expert. Using his or her domain knowledge and intuition, a domain expert can anticipate certain characterising aspects of a process without having to do any extensive calculations merely with a particular, simplified physical model in mind. In the case of a manufacturing process, this model describes the interaction between a tool and a workpiece due to the axial movement of the latter and abstracts away any other effects.

In order to make access to time series data as seamless as possible, it is necessary to make as many of the intuitively accessible aspects (overall shape and expected system dynamic per physical quantity) usable and minimise the need for others (exact measurement values, durations, points in time).

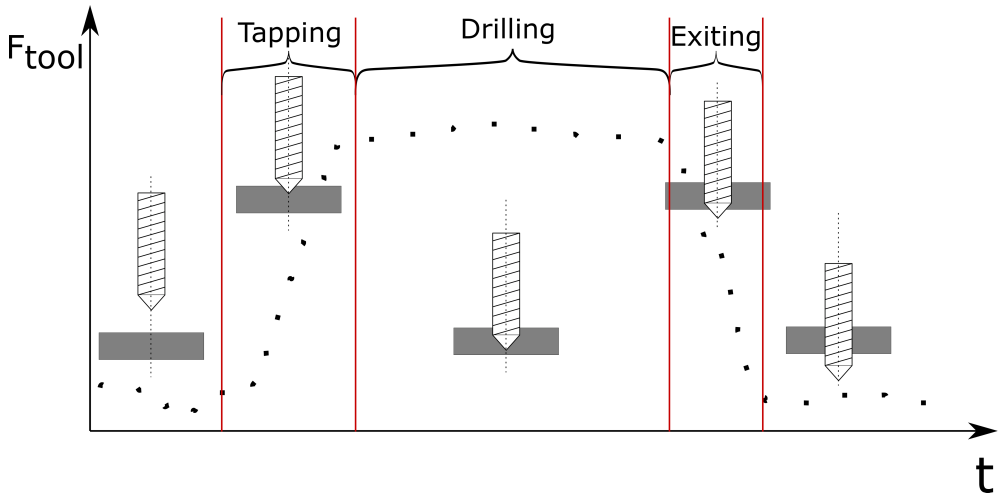


Figure 6.9 Expected course of force measurements for a drilling process.

6.5.1 Semantic Modelling for Time Series Data

A comparison between expected shapes and real data (Figure 6.10), makes apparent differences obvious. Even though the general shape somehow resembles what would be expected, raw data tends to contain much more signals than just the ones considered by humans when expectations are formulated. As always, reality tends to be much more complicated then we would expect. This needs to be addressed before human notions can be used to access time series data automatically.

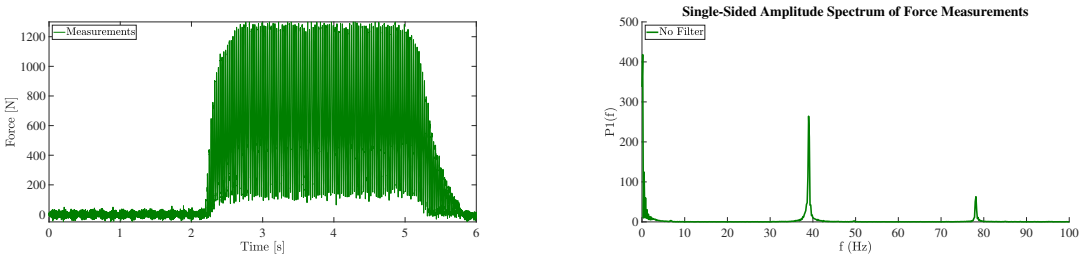


Figure 6.10 Force measurement data sampled at 2kHz (left) and discrete frequency spectrum of the force measurement data (right).

According to the theory behind Fourier analysis, a general function in the time domain can be approximated by sums of trigonometric functions. Continuous Fourier transformation of time series data produces a continuous function of frequency, the frequency distribution (Equation 6.4a).

Similarly, this transformation can be defined for discrete time series, such as those that are considered in this work (Equation 6.4b). They, in turn, produce discrete frequency distributions (Figure 6.10).

$$S(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-i2\pi ft} dt \tag{6.4a}$$

$$S(k) = \sum_{n=0}^{k} x_n \cdot e^{-\frac{i2\pi}{N} kn} \tag{6.4b}$$

Under this perspective, also the measurement depicted in Figure 6.10 can be seen as the sum of trigonometric functions. Shapes, as defined by domain experts, implicitly correspond to only a subset of those functions (frequencies). Based on the physical quantity and the investigated phenomenon (*system*), changes in the measured signal that correspond to the mental model of domain experts can be expected to be more or less dynamic. Consequently, when shapes are defined, it is necessary also to define the frequency domain that those shapes correspond to. This can be done through time constants, which define transfer functions, and therefore are characteristic for any given system. A time constant corresponds to precisely one frequency and can alternatively be interpreted as a measure for the minimum duration that a given interval has to have before it is considered for classification based on its shape.

This idea can be used to determine filters for measurements. The relationship between time constants (T_u) and minimum durations (T_{min}) are given in Equations 6.5. As already pointed out, any time constant corresponds to exactly one harmonic oscillation of a particular frequency (Equation 6.5a, 6.5b). Given this, the minimum duration of intervals is defined using the Nyquist–Shannon sampling theorem, which defines the minimum sampling frequency necessary to permit information loss while converting a continuous-time signal with a finite bandwidth into a sequence of discrete samples. In this case, however, the theorem is used in the opposite direction. For a given frequency, the minimum frequency according to the Nyquist-Shannon theorem can also be seen as the threshold for a maximum duration ($f = \frac{1}{T}$) that phenomena can have before they are lost due to under-sampling (Equation 6.5c). Using this criterium, given the sampling rate of the respective sensor f_{sample}, minimum durations can be translated into a filter parameter (N_{window}, Equation 6.5d), which can be used to define a moving average filter (Equation 6.5e).

$$x(t) = x_0 \cdot sin(\omega t + \varphi_0) \tag{6.5a}$$

$$x(t) = x_0 \cdot sin(\frac{2 \cdot \pi}{T_g} t + \varphi_0) \tag{6.5b}$$

$$f_{min} = \frac{1}{T_{min}} = 2 \cdot \frac{2 \cdot \pi}{T_g} \tag{6.5c}$$

$$N_{window} = \frac{T_{min}}{f_{sample}} \tag{6.5d}$$

$$\bar{x}(n) = \frac{\sum_{i=n_{start}}^{n_{end}} x(i)}{N_{window}} \text{ with } n_{start} = n - (N_{window} - 1) \tag{6.5e}$$

The exact value of a time constant can be determined based on domain knowledge. Two ways shall be illustrated further based on the running DRILLINGPROCESS -example. Respective results for both approaches can be seen in Figure 6.12 and Figure 6.13.

1. System Identification: In the case that experimental data is available, using a step function response, the tangent of inflexion can be constructed. Based on that, the equivalent dead time (T_u) and the equivalent time constant (T_g) can be determined. In Figure 6.11a, this is illustrated further. For the example measurement, through this approach a time constant value of $T_g \approx 0.2$ can be estimated.

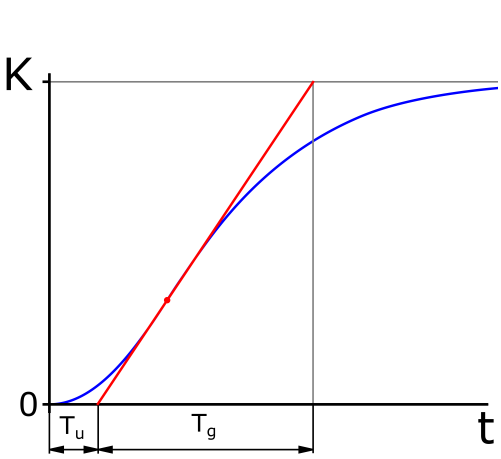
2. Analytical Estimation: Using knowledge about the domain, simple calculations can be done to estimate the expected system dynamic. The schema depicted in Figure 6.11b shows the required parameters to estimate the time constant. Tool dimensions (x_t) and process parameters (v_f) are used to determine the expected duration of the tipping phase, which is a good approximation for the time constant. For the example measurement, a time constant value of $T_g \approx 0.33$ can be estimated (Equation 6.6).

$$T_g \approx \frac{x_t}{v_f} = \frac{3mm}{10\frac{mm}{s}} = 0.33s \tag{6.6}$$

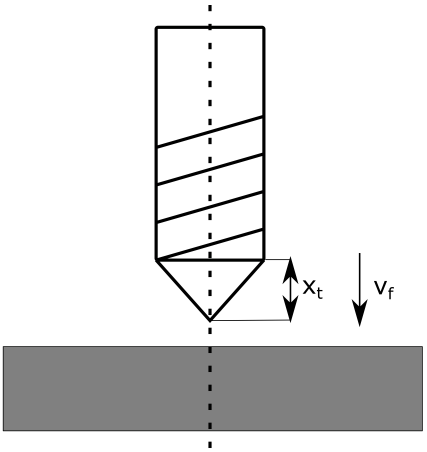
As can be seen, the time constant that was determined through system identification turned out to be too small. For the analytically determined one, however, a somewhat satisfying result was achieved. In both cases, the target frequency was relatively high. The spectrum of the signal shows that also much higher time constants would be feasible, and therefore, much more selective filtering can be conducted before frequencies connected with the target phenomenon start to be affected.

6.5.2 Use Cases

As established, the manufacturing domain requires an ontology language that can capture interval aggregate functions such as moving average together with various metric constructs. In this section, a

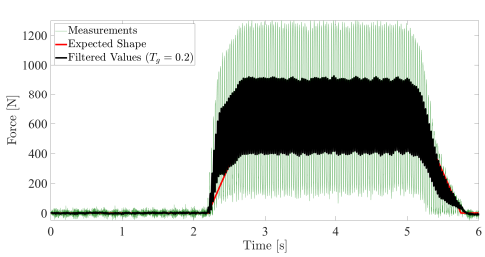


(a) System Identification from a step function response (blue) using the tangent of inflexion (red).



(b) Schematic Illustration used for analytically determining the time constant for a drilling process.

Figure 6.11 Illustrations regarding determination methods for time constants. System identification (left) and analytical determination (right).



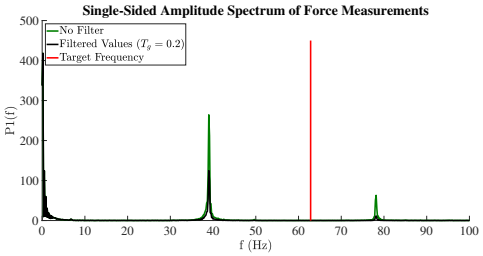


Figure 6.12 Filtered data (compared with expected shapes for time constants determined based on system identification (left) and respective spectrum (right).

new rule language, which was developed together with colleagues from the Free University of Bozen-Bolzano and Birkbeck, University of London, will be introduced. The authors work in this respect was the formulation and definition of the use cases as well as the validation of logical interpretations of the concepts in order to make sure they actually reflect the intended meaning. This language was developed with the goal to makes it possible to consider aggregates over temporal windows in the context of OBDA. For a full description of the language, refer to [30]

In [36, 105, 109], such aggregates were possible, but only allowed in queries, and so again an IT intermediary may be needed to help the user; [12] introduced description logics with aggregation features and showed that reasoning with them is often undecidable.

We are looking for particular sequences of *shapes* such as an interval (tapping) when the force is increasing followed by an interval (drilling) when the force is stable and then by an interval (exiting) when the force is decreasing. The duration of each of these intervals and the force values may vary widely due to different combinations of workpiece materials, tools, process parameters and tool wear.

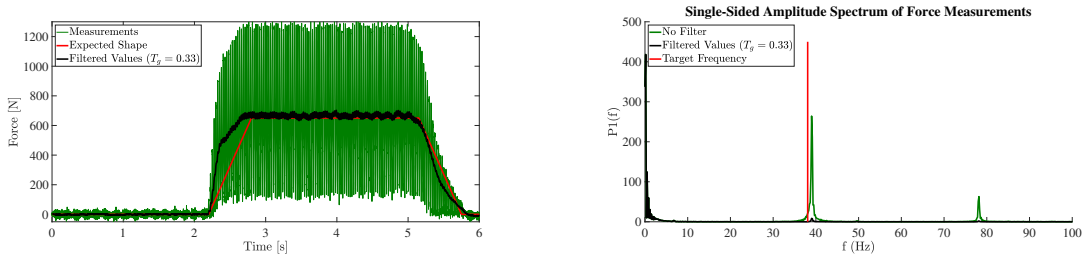


Figure 6.13 Filtered data (compared with expected shapes for time constants determined based on analytical estimation (left) and respective spectrum (right).

The proposed framework contains unary predicates for representing events over temporal intervals and binary predicates for capturing numerical sensor measurements over temporal intervals and also the results of aggregation. Thus, *DSLD* is essentially two-dimensional, with a temporal dimension comprising intervals and a (measurement) value dimension $\mathbb{R}$.

As *interval* we understand a set $(\mathbb{R},<)$ of real numbers. More precisely, an *interval*, ι, is any nonempty subset of $\mathbb{R}$ of the form $\langle t_1,t_2\rangle$, where $t_1,t_2 \in \mathbb{R}\cup\{-\infty,\infty\}$, '$\langle$' is '(' or '[' and '$\rangle$' is ')' or ']'. Note that we admit *punctual* intervals of the form $[t,t]$. We denote by $\mathsf{int}_{\mathbb{R}}$ the set of all intervals over $\mathbb{R}$ and by $|\iota|$ the *length* of an interval ι. We use the variables $\boldsymbol{x},\boldsymbol{y},\ldots$ for intervals.

Sensor measurements are deemed to be real numbers. We write $R(\iota,v)$ to say that $v \in \mathbb{R}$ is the *value* measured by sensor R over the interval $\iota \in \mathsf{int}_{\mathbb{R}}$, and we write $A(\iota)$ to say that event A occurs in the interval ι. We use the variables $x,y,\ldots$ for values.

Drilling Process

A drilling process can be captured by the following program:

$$\mathit{NormalDrilling}(\boldsymbol{x}) \leftarrow \mathit{Tapping}(\boldsymbol{x}_1),\mathit{Drilling}(\boldsymbol{x}_2),\mathit{Exiting}(\boldsymbol{x}_3),\\ \boldsymbol{x}_1\,\mathsf{A}\,\boldsymbol{x}_2,\ \boldsymbol{x}_2\,\mathsf{A}\,\boldsymbol{x}_3,\ \boldsymbol{x}\,\mathsf{is}\,(\boldsymbol{x}_1 \uplus \boldsymbol{x}_2 \uplus \boldsymbol{x}_3)$$

Here, $\boldsymbol{x}$ and the $\boldsymbol{x}_i$ are intervals over the real numbers, $\uplus$ returns the union of connected intervals in case it is also a (connected) interval, and the predicates $\mathit{Tapping}(\boldsymbol{x}_1)$, $\mathit{Drilling}(\boldsymbol{x}_2)$ and $\mathit{Exiting}(\boldsymbol{x}_3)$ involve complex interval aggregation to smooth out the fluctuating measurements of force and ensure that it is increasing, stable and decreasing, respectively.

This basic notion can be expressed in a more detailed way using the following rules.

$$
\begin{aligned}
\mathit{AvgForce}(\boldsymbol{x},y) &\leftarrow \mathit{wavg}\{(\boldsymbol{x},y) \mid \mathit{Force}(\boldsymbol{x},y)\},\\
\mathit{AvgForce}'(\boldsymbol{x},y) &\leftarrow \mathit{wavg}\{(\boldsymbol{x},y) \mid \mathit{Force}(\boldsymbol{x},y)\}, D^{[,)}_{0,13s}(\boldsymbol{x}),\\
\mathit{ForceDelta}(\boldsymbol{x},z) &\leftarrow \mathit{AvgForce}'(\boldsymbol{x},y), \mathit{AvgForce}'(\boldsymbol{x}',y'), \boldsymbol{x}\,\mathsf{A}\,\boldsymbol{x}', z\,\mathsf{is}\,(y-y'),\\
\mathit{IncForce}(\boldsymbol{x}) &\leftarrow \mathit{coalesce}\{\boldsymbol{x} \mid \mathit{ForceDelta}(\boldsymbol{x},d), d > 0.1\},\\
\mathit{ConstForce}(\boldsymbol{x}) &\leftarrow \mathit{coalesce}\{\boldsymbol{x} \mid \mathit{ForceDelta}(\boldsymbol{x},d), |d| \leq 0.1\},\\
\mathit{DecForce}(\boldsymbol{x}) &\leftarrow \mathit{coalesce}\{\boldsymbol{x} \mid \mathit{ForceDelta}(\boldsymbol{x},d), d < -0.1\},\\
\mathit{SimpleDrilling}(\boldsymbol{x}) &\leftarrow \mathit{IncForce}(\boldsymbol{x}_1), \mathit{max}\{(\boldsymbol{x}_1,y_1) \mid \mathit{AvgForce}(\boldsymbol{x}_1,y_1)\},\ \boldsymbol{x}_1\,\mathsf{A}\,\boldsymbol{x}_2,\\
&\quad \mathit{ConstForce}(\boldsymbol{x}_2), \mathit{max}\{(\boldsymbol{x}_2,y_2) \mid \mathit{AvgForce}(\boldsymbol{x}_2,y_2)\},\ \boldsymbol{x}_2\,\mathsf{A}\,\boldsymbol{x}_3,\\
&\quad \mathit{DecForce}(\boldsymbol{x}_3), \mathit{max}\{(\boldsymbol{x}_3,y_3) \mid \mathit{AvgForce}(\boldsymbol{x}_3,y_3)\},\\
&\quad |y_1-y_2| < y_2 \times 0.05,\ |y_2-y_3| < y_3 \times 0.05,\\
&\quad \boldsymbol{x}\,\mathsf{is}\,(\boldsymbol{x}_1 \uplus \boldsymbol{x}_2 \uplus \boldsymbol{x}_3).
\end{aligned}
$$

In the first rule, the concept *AvgForce* is defined, using an aggregation function ($\mathit{wavg}\{(\boldsymbol{x},y)$) on the time series $\mathit{Force}(\boldsymbol{x},y)\}$ over the interval $\boldsymbol{x}$. Furthermore, a filtered signal $\mathit{AvgForce}'(\boldsymbol{x},y)$ can be defined using a **discretization scheme** $D^{[,)}_{0,13s}(\boldsymbol{x})$.

Based on this filtered signal, the rule $\mathit{ForceDelta}(\boldsymbol{x},z)$, which compares values of temporal subsequent functions, can be used to define the concepts $\mathit{IncForce}(\boldsymbol{x})$, $\mathit{ConstForce}(\boldsymbol{x})$, $\mathit{DecForce}(\boldsymbol{x})$. These intervals, if found in the correct order, are combined to form the $\mathit{SimpleDrilling}(\boldsymbol{x})$ concept. This order is expressed using Allen's relations.

Smooth Drilling

Drilling processes can be further classified based on other characteristics within the force measurements. Throughout such a drilling process, however, anomalies can occur, the simplest of which is the presence of significant peaks which might come, for example, from material inhomogeneities. In that sense, a *smooth drilling process* can be distinguished by evaluation of the maximum and minimum values within the simple drilling event.

$$
\begin{aligned}
\mathit{Drilling}(\boldsymbol{x}) &\leftarrow \mathit{ConstForce}(\boldsymbol{x}),\\
\mathit{SmoothDrilling}(\boldsymbol{x}) &\leftarrow \mathit{SimpleDrilling}(\boldsymbol{x}),\ \mathit{Drilling}(\boldsymbol{x}_1),\ \boldsymbol{x}_1 \subseteq \boldsymbol{x},\ \mathit{AvgForce}(\boldsymbol{x}_1,y),\\
&\quad \mathit{max}\{(\boldsymbol{x}_1,y_1) \mid \mathit{AvgForce}(\boldsymbol{x}_1,y_1)\},\ y_1 - y \leq 0.1 \times y,\\
&\quad \mathit{min}\{(\boldsymbol{x}_1,y_2) \mid \mathit{AvgForce}(\boldsymbol{x}_1,y_2)\},\ y - y_2 \leq 0.1 \times y.
\end{aligned}
$$

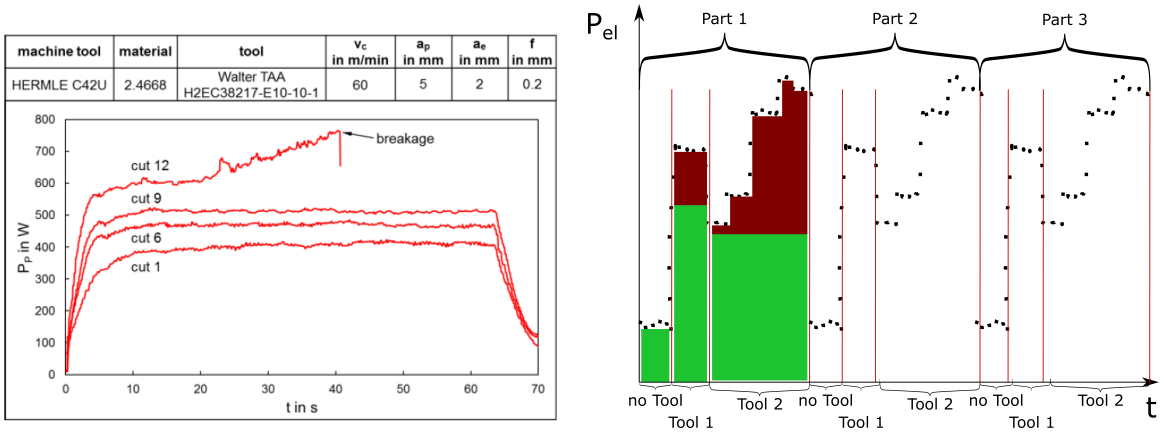
In this rule, *coalesce* represents the aggregation function coalescing, which computes the maximal intervals with the same value. Similarly, *max*, *min* and *wavg* are aggregation functions computing the maximum, minimum and weighted average values over an interval, respectively.

Unstable Drilling

The event *unstable drilling process* can be classified based on the standard deviation of the force readings. Even though they might look similar based on their average process forces in each phase, unstable processes will have significantly higher standard deviations then stable ones. Again, the threshold that would be used here has to be defined based on domain knowledge.

$$SmoothDrilling(\boldsymbol{x}) \leftarrow SimpleDrilling(\boldsymbol{x}),\ Drilling(\boldsymbol{x}_1),\ \boldsymbol{x}_1 \subseteq \boldsymbol{x},\ AvgForce(\boldsymbol{x}_1, y_1),$$
$$sdev\{(\boldsymbol{x}_1, y_2) \mid Force(\boldsymbol{x}_1, y_2)\},\ |y_2 - y_1| \leq 0.1 \times y_1,$$

Similar to the rule defined above, *sdev* represents an aggregation function computing the standard deviation over an interval.



(a) Electrical power readings for a sequence of similar cuts until tool break occurred [81].

(b) Total electrical power input (P_{el}) for a production machine while manufacturing three similar parts.

Figure 6.14 Filtered data (compared with expected shapes for time constants determined based on analytical estimation (left) and respective spectrum (right)

Tool Break

Tool breaks happen regularly in industrial production processes. Example power readings for such an event are depicted in Fig. 6.14a. In this example, power is directly related to apparent process forces. As a consequence, force readings would differ from the displayed power readings only by a constant conversion factor. To identify tool breaks, we simply find any patterns that match the concept *simple drilling*, but are shorter then some expected duration:

$$ToolBreak(\boldsymbol{x}) \leftarrow SimpleDrilling(\boldsymbol{x}),\ ExpectedDrillingTime(\boldsymbol{x}, y),\ |\boldsymbol{x}| \leq y,$$

This expected duration is the duration of a drilling process (t_{exp}) if no anomalies (like a tool break) appear. The exact value for this can be determined based on process parameters as it is done in Computer Aided Manufacturing (CAM) tools.

Energy per Tool

Closely related to tool wear is the remaining tool lifetime. One way to determine this measure from sensor data is the amount of accumulated electrical energy that was used to drive a given tool. To illustrate, consider Fig. 6.14b showing power measurements for a production machine. Within the observed window, three similar parts (having the same geometric features) are manufactured. In order to do so, two different tools are used. To calculate the energy per tool (E_{tool}), the power measurements (P_{tool}) need to be integrated over all past intervals (i) in which the respective tool was active (6.7a). Before doing so, however, it is necessary to deduct the base load (P_{base}) of the machine (6.7b). A machines base load is defined as its (electrical) power demand without material removal due to auxiliary systems or power loss in bearings.

$$E_{tool} = \sum_{i}^{k} \int_{s_i}^{t_i} P_{tool}(t)dt \tag{6.7a}$$

$$P_{tool}(t) = P_{el}(t) - P_{base}(t) \tag{6.7b}$$

These notions can be modelled using the following rules:

$$\begin{aligned}
Pbase(\boldsymbol{x}, v) &\leftarrow SimpleDrilling(\boldsymbol{y}), \boldsymbol{x} \,\mathsf{is}\, lext_{-10s}(\boldsymbol{y}), wavg\{(\boldsymbol{x}, v) \mid Power(\boldsymbol{x}, v)\},\\
Ptool(\boldsymbol{x}, u) &\leftarrow wavg\{(\boldsymbol{x}, v) \mid Power(\boldsymbol{x}, v)\}, Pbase(\boldsymbol{x}', v'), \boldsymbol{x} \subseteq \boldsymbol{x}', u \,\mathsf{is}\, v - v',\\
Etool(\boldsymbol{x}, v) &\leftarrow int\{(\boldsymbol{x}, v) \mid Ptool(\boldsymbol{x}, v)\}, SimpleDrilling(\boldsymbol{x}),
\end{aligned}$$

where, for a functional relation R, aggregation functional $int(R)$ is defined as

$$int(R) = \left\{ \left(\iota, \int_{\iota} f_R(x)\,dx\right) \mid \iota \in \mathsf{int}_{\mathbb{R}} \text{ is bounded and } \iota \subseteq dom\, R \right\}.$$

where $f_R(x)$ is the function corresponding to R, and is undefined otherwise.

6.5.3 System Evaluation

The defined concepts can be translated to SQL queries using an algorithm described in Brandt et al. [30]. These where evaluated using 2.6 GB of real data from the described database (all available force and energy measurements at the moment of writing this thesis). Experiments ran on an AWS server with an Intel Xeon Platinum-8175 processor having 8 logical cores at 2.5 GHz and 64 GB of RAM. The SQL queries were executed on Apache Spark 2.4.0.

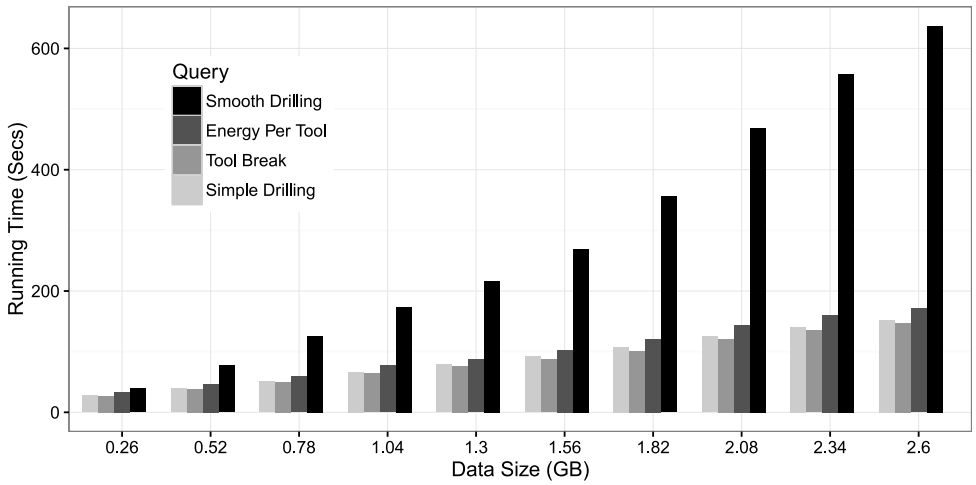


Figure 6.15 Running times of the drilling queries.

Figure 6.15 shows the times for the manufacturing rig use case. These results show that the execution times scale linearly over monotonically increasing data. Also, while all queries stay in reasonable execution time boundaries (maximum Ĩ0 min), especially the rather dramatic development of the execution time for the SMOOTHDRILLING concept has to be pointed out. Reasons for this need to be investigated further.

Chapter 7

Conclusion

This thesis tried to investigate the potential and limitations of OBDA to overcome current problems concerning data science in the manufacturing domain. In this last chapter, the results of the research conducted will be summed up, and questions formulated at the beginning of this document are answered. Furthermore, some suggestions on future research topics will be given.

7.1 Discussion

Using the results of this research, research questions formulated initially can be answered.

7.1.1 Research Question 1

Regarding the first research question (RQ 1), it was expected that, for the given example use case, more than 60% of the total time spent was due to activities classified as data access (Hypothesis 1). In Chapter 4, such an example use case was introduced. Using time records from this cooperative development effort, it could be shown that, at least in this instance, the claims made by Kharlamov et al. [100] are supported. According to the calculations, roughly 66% of the total time spent on implementing the described prototype can be considered as data access, **which supports Hypothesis 1**.

For this result, a few limitations need to be taken into consideration. First, this is just another example, and it would not be arguable to generalise this result in any way. In order to investigate the general role of data access, further research would have to be conducted. Furthermore, the data used was not as detailed as necessary, which led to the necessity to make some rather strong assumptions. Nevertheless, even though those assumptions are a source of considerable uncertainty, it is very likely that of all the defined process phases, data access would still end up being the most important one in terms of time spent.

7.1.2 Research Question 2

In respect to the second research question (RQ 2), Hypothesis 2 was formulated, which claims that integration of heterogeneous data sources and time series data are main requirements towards data science within the manufacturing domain. Furthermore, the claim that limited IT knowledge is a problem. To investigate this, a literature review was conducted. For this review, two works ([83, 52]) could be used as a starting point. Based on this, an attempt to review articles on data mining in the manufacturing domain that were not covered by those reviews was made. Those articles were separated into non-temporal and temporal scenarios. Just the number of articles in each group (9 non-temporal and 36 temporal) supports the supposed importance of temporal data in this domain. A similar point can be made in respect to the topic of data integration. Hardly any of the reviewed articles were concerned with data that would in realistic conditions be stored in a single data source. The majority of scenarios combines data from different systems such as MES, ERP or tool management systems.

Adding to the results from the literature, also two application scenarios where introduced. While access to temporal data was an integral feature for both of them, data integration was not directly considered a problem. This, however, is due to the research-oriented nature of the considered projects. This enabled us to create a system from scratch without having to build upon existing data sources as it would be the case in real-world applications.

Regarding the claim that the lack of relevant knowledge is a limiting factor, some support could be found. This, however, is not a statement that holds only in the manufacturing domain. The problem of insufficient IT knowledge concerning data science becomes problematic as soon as the application domain is anyone but computer science. More precisely, as soon as the problems that have to be investigated are others then databases, the knowledge required to access the data is different from that which is required to formulate questions and interpret results. Consequently, for manufacturing just as for virtually any other domain of discourse, not lack of knowledge can be considered the problem, but rather the lack of sufficient tools that give access to data which is necessary to answer domain-specific questions through data analysis.

All in all, Hypothesis 2 was not falsified. As additional requirements, aggregation functions could be identified. Being able to handle this kind of functions efficiently, both over temporal and non-temporal domains, is required for most data science tasks and therefore needs to be taken into consideration explicitly.

7.1.3 Research Question 3

In order to answer research question 3, first another literature review was conducted. This review, which can be found in Chapter 3, yielded encouraging results in respect to the potential of OBDA. In particular, the fact that already some research was done in the direction of temporal OBDA indicated that this technology might be suitable for tackling at least some of the requirements from the manufacturing domain. In order to further test Hypothesis 3, a PoC was developed. Semi-structured,

qualitative interviews with stakeholders further indicated, that this technology indeed could ease common problems in the particular application scenario. Also, other application scenarios where identified by the interviewed stakeholders. Consequently, a prototype was developed for the second application scenario. Apart from a more elaborate, domain-specific, static ontology, an attempt was made to improve the ability of OBDA to handle temporal data.

To this end, existing modelling approaches to approach temporal data in an OBDA context where identified to not be fitting for data science applications. Data science, by definition, is exploratory and often domain experts cannot be expected to be able to formulate temporal concepts using the parameters that, for example, DatalogMTL would provide them with. Only in sporadic cases it is possible to name the exact value or time limits that define a concept. As a consequence, an alternative modelling approach (which is defined in Chapter 6) was chosen. Using this, domain experts can use intuitively accessible parameters such as the shape of measurement and the general time domain in which described phenomena are expected to appear to define concepts of interest. Together with colleagues, a formal language suitable for capturing these concepts was defined. In conclusion, **Hypothesis 3 was supported** by the findings of this thesis. As expected, temporal data is necessary for any OBDA solution to be suitable for the manufacturing domain. Remarkably, aggregates over temporal windows are not only helpful in defining relevant concepts (i.e. ENERGYPERTOOL), but also for filtering real-world time series data.

7.1.4 Research Question 4

Finally, regarding research question 4, example queries from the second application scenario where used to evaluate the prototype. In the case of non-temporal queries, the formulated **Hypothesis 4 had to be falsified**. This is because the chosen OBDA framework currently does not support aggregations functions. Therefore, while the queries are arguably much simpler then in a conventional setting, the results from OBDA queries need to be post-processed while those from conventional SQL queries can be used directly. In the case of temporal data, current frameworks cannot be used to express all necessary notions needed to facilitate direct access to time series data. Again, aggregates, but over temporal windows, would be required to make OBDA truly useful as a technology to facilitate data access in the manufacturing domain. Consequently, the first steps towards the development of a corresponding rule language were made. Based on this work, some further research topics can be identified.

7.2 Further Research Topics

Based on investigations made in this thesis, some further research topics where identified. These are described in the following sections.

7.2.1 Enabling cross-organisation Data Exchange

The application scenarios presented in this thesis only required a limited exchange of data across organisational boundaries. In the first application scenario (Chapter 4), data exchange between Infineon Austria GmbH and TU Vienna was necessary. In the second scenario (Chapter 5), data were analysed by the same organisation that produced it. In reality, however, for data science use cases to unfold their full potential, close cooperation between organisations (i.e. manufacturers, contractors, logistics service providers or research institutions) is required.

Imagine, for example, the initially formulated vision of a shared manufacturing process dataset containing process parameters, quality results and other factors for manufacturing scenarios covering any combination of technology, machine, tool and material that was ever investigated by anybody on the planet. Manufacturers could consult this dataset to find optimal process parameters for an exotic material that they did not use before. Data scientists could query this dataset in order to find patterns, identify technological developments and define areas in which further research might be needed. Integration of manufacturing process data could also reduce the auditing efforts required between company partners.

No single organisation in the world can be expected to have such a full dataset. Nor does it seem to be a desirable scenario either. Collectively, however, the data generated by all manufacturers around the globe might get reasonably close to this ideal if it was combined. Several questions regarding the realisation of such a shared dataset in the manufacturing domain arise. One of those questions, regarding technological solutions to facilitate this data sharing (which is an instance of the data integration problem) was partly investigated in this thesis. The application of OBDA was shown to be a suitable tool to solve (intra-organisational) data integration problems. Existing approaches can be extended in order to be able to reflect also time series data. The idea, that similar problems, but on a larger scale (crossing organisational borders), could be solved through virtual knowledge graphs seems reasonable. Ontologies are a means to combine knowledge which does not necessarily need to be agreed upon by all users of a system. As a consequence of their original purpose, they are designed to mediate inconsistent pieces of knowledge and still make it possible for them to be combined. The application of OBDA for cross-organisational data sharing is, therefore, an interesting field of further research.

Apart from technological questions regarding the facilitation of data sharing, for such a scenario to be realistic, open issues regarding competition and incentives arise. Sharing process knowledge in the form of raw data is a risk for any business. This is especially true when, as it is the case in any data science scenario, the amount of knowledge hidden within that data cannot be known beforehand. On the other hand, giving others access to (parts of) the data that is generated throughout manufacturing might end up being an attractive source of additional income for many companies. To realise this, technologies to simplify the value exchange to compensate data generators for their efforts and risks need to be introduced.

7.2.2 Investigation of new Language for Time Series Logs

The language that was introduced in Section 6.5.2 still is in a very early stage. Currently, no formal proofs regarding its complexity are available. Furthermore, currently, no frameworks or tools exist that would make the use of such a language possible. The development and experimental investigation of this language could, however, pave the way for broader, industrial application.

7.2.3 Handling of Functions

When the first example for temporal data analysis tasks (energy per tool in Section 6.5.2) was introduced, the total electrical power input of the machine P_{el} and the machines base load P_{base} where named as required inputs. Often, however, the base load is not explicitly measured. Therefore, assumptions must be made. These assumptions represent pieces of knowledge and should, consequently, also be captured within an ontology.

An example piece of knowledge, which could be used to deduct the base load power demand without an explicit measurement could, for example, be the relationship between base load and spindle rotation speed as it is depicted in Figure 7.1. The general shape of this function can be assumed to be similar for all machines of the same type (or rather the same spindle type). The coefficients of the underlying function, however, might be different. Due to energy dissipation in the spindles bearings, which increases for higher rotation speeds, the power demand increases for higher rotation speeds. The plateau in the graph is because such spindles are optimised for certain rotation speed ranges.

Access to such a piece of knowledge (which assigns to each value of n a respective value of P_{base}) makes it possible to substitute the missing power measurement. The concrete way to capture this dependency is yet to be defined.

Similar to what was previously described, another motivating scenario for improved handling of functions is depicted in Figure 7.2. It depicts fictional temperature readings from inside an oven. At 4 am, someone decides to switch it on in order to bake buns. To monitor the temperature increase, every hour a manual temperature measurement is carried out. Furthermore, a sensor is deployed to the said oven in order to measure its thermal power input. This sensor has a much higher sampling rate than manual measurement.

Suppose now, that the operator of the oven would like to know how much of the increased power input follows from the temperature increase alone. This would require to deduct some (now unknown) fraction of the total heat input ($\dot{Q}_{oven}$) before integrating in order to, for example, then get the total energy consumed for the temperature increase. This unknown fraction can be considered to be heat loss, which can not be measured directly. Luckily, if we assume only conductive heat transport, the

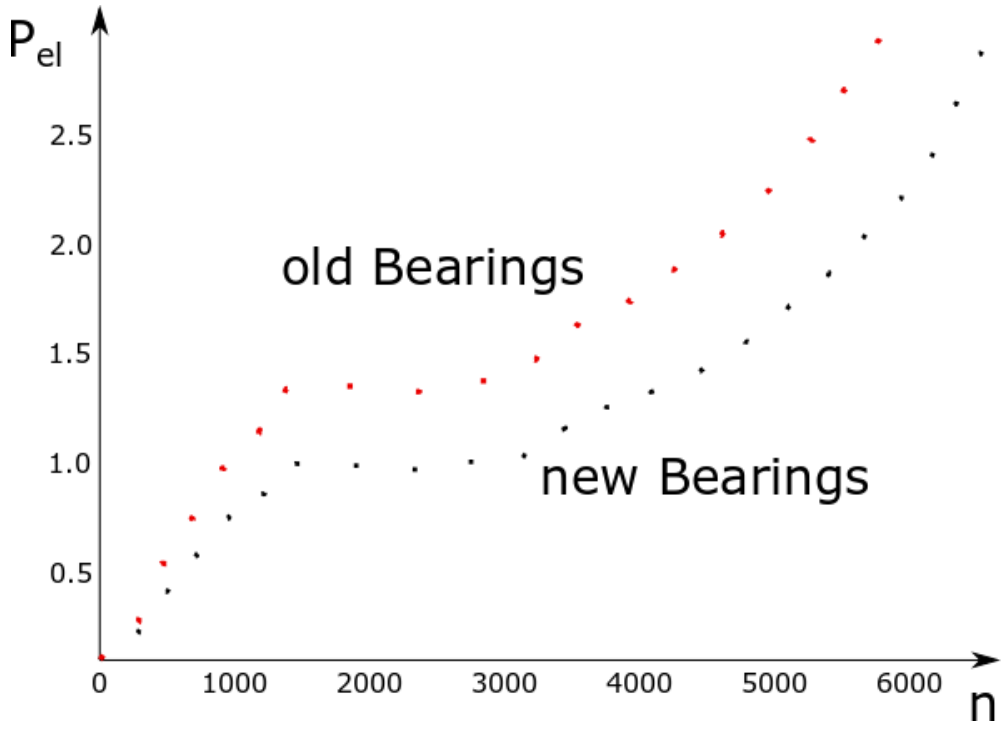


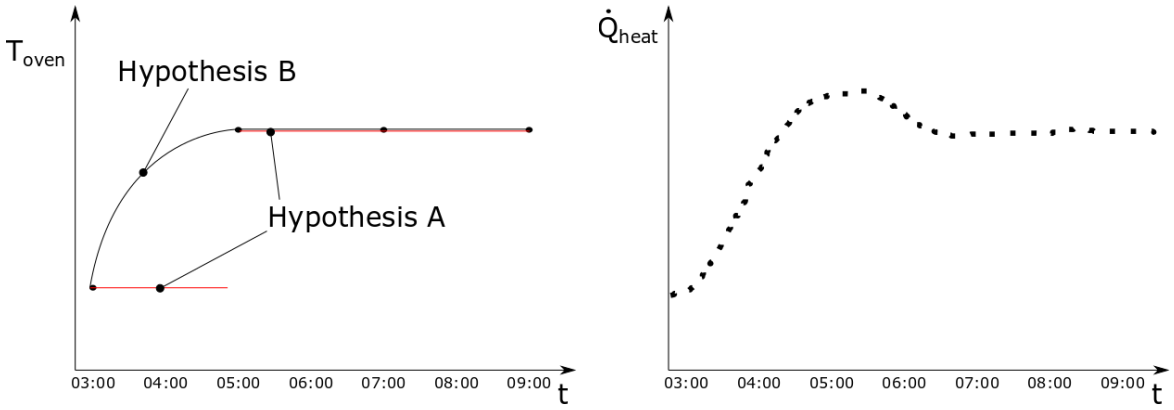
Figure 7.1 Dependency between main spindle power demand (P_{el}) and rotation speed (n).

relationship between the temperature of the oven (T_{oven}) and the ambient temperature ($T_{ambient}$, which can be assumed to be constant) is known (Equation 7.1b).

$$\dot{Q}_{increase}(t) = \dot{Q}_{oven}(t) - \dot{Q}_{loss}(t) \tag{7.1a}$$

$$\dot{Q}_{loss}(t) = \theta \cdot A \cdot \frac{T_{oven}(t) - T_{ambient}}{d} \tag{7.1b}$$

Apart from the fact, that this particular function could (and arguably should) be made available through an ontology, in this particular scenario, we face another problem. Even though the necessary input is measured, the sampling rate is much lower than that of the heat power measurement. This results, depending on the assumption made for points that are within intervals and not explicitly measured, in potentially high errors. Consider for example the assumption made in Hypothesis A, where the temperature is expected to be constant in any interval and has the value of the previous measurement. There is a significant difference between this assumption and what is known about the behaviour of such systems (Hypothesis B). The knowledge behind Hypothesis B is reflected in the



(a) Internal oven temperature (T_{oven}) according to sensor readings (Hypothesis A) and according to domain knowledge (Hypothesis B).

(b) After the oven is switched on, heat input increases until the target temperature is reached.

Figure 7.2 Two measurements from a baking oven while heating up. The sampling frequency of the internal temperature is much lower then that of the oven power input ($\dot{Q}_{heat}$).

form of differential equations which describe the temperature change within a closed thermodynamic system which is subject to heat exchange with external sources(7.2a-7.2d).

$$m \cdot c_p \cdot \frac{dT_{oven}}{dt} = \dot{Q}_{oven}(t) = \dot{Q}_{heating}(t) - \dot{Q}_{loss}(t) \tag{7.2a}$$

$$m \cdot c_p \cdot \frac{dT_{oven}}{dt} = \dot{Q}_{heating}(t) - \theta \cdot A \cdot \frac{T_{oven} - T_{ambient}}{d} \tag{7.2b}$$

$$\frac{dT_{oven}}{dt} + \frac{\theta \cdot A}{m \cdot c_p \cdot d} \cdot T_{oven} = \dot{Q}_{heating}(t) + \frac{\theta \cdot A \cdot T_{ambient}}{m \cdot c_p \cdot d} = 0 \tag{7.2c}$$

$$T_{oven} = e^{\frac{\theta \cdot A}{m \cdot c_p \cdot d} \cdot t} + T^p_{oven} \tag{7.2d}$$

If it were possible to use knowledge about system behaviour, this knowledge could also be used to improve estimations within intervals which typically come from interpolations. Domain knowledge can improve quality by providing correct interpolation functions.

Glossary

API In computer programming, an application programming interface (API) is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

BaMa As part of the research project Balanced Manufacturing (BaMa), software solutions were developed that enable companies to combine the success factors of energy, time, costs and quality in production and operational planning. These solutions are based on a novel, scientific method for the production of holistic, virtual images of production plants as well as their optimisation. The developed methods allow operational management decisions to be operationalised quickly without having to buy new hardware. A use case from this project was used to illustrate the application of OBDA in the manufacturing domain.

BOM The Bill of Material (BOM) lists raw materials and (sub-) assemblies and their respective quantities required for manufacturing products..

CFRP Carbon- Firbe Reinforced Polymer (CFRP), a composite material used, for example, in the aviatiln industry.

CSV In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

DBMS Database Management System (DBMS) are the basic software that support the management of data sored on a computer (database) [1].

DL Description Logic (DL), a very active research area in logic-based knowledge representation and reasoning that goes back to the late 1980s and that has a wide range of applications in knowledge-intensive information systems [18].

ERM In the context of relational databases, an Entity- Relationship- Model (ERM) is used to describe data as entities, relationships and attributes.

ERP Enterprise Resource Planning (ERP) is defined as an integrated computer-based system that manages internal and external organizational resources. These resources include tangible assets, financial resources, materials and human resources.

ETL In computing, extract, transform, load (ETL) refers to a process in database usage and especially in data warehousin..

FIR Finite Impulse Response (FIR) is a class of filters which only consider past inputs and therefore produce finite responses when processing an impulse.

FOL First-order logic is a collection of formal systems. It uses quantified variables such as "for all" and "there exists". The adjective "first-order" distinguishes first-order logic from higher-order logic in which there are predicates having predicates or functions as arguments, or in which one or both of predicate quantifiers or function quantifiers are permitted [20].

IFT Institute for Production Engineering and Laser Technology is part of the faculty for mechanical engineering at the Technical University Vienna.

INF Infineon Austria AG (INF). Within the BaMa project, simulation-based optimisation was used to optimise the energy supply for the air conditioning of clean rooms. After identification of the essential parts of the plant (chillers, recooling plants, thermal energy networks) the corresponding monitoring data were analysed and prepared for model identification.

IRI The Internationalized Resource Identifier (IRI) – is an internet protocol standard which extends the ASCII characters subset of the Uniform Resource Identifier (URI) protocol. While URIs are limited to a subset of the ASCII character set, IRIs may contain characters from the Universal Character Set (Unicode/ISO 10646), including Chinese or Japanese kanji, Korean, Cyrillic characters, and so forth..

IT Information technology (IT) is the use of computers to store, retrieve, transmit, and manipulate data, or information.

JDBC A database interface provided by the Java platform, can be used to connect to DBMS from different vendors..

JSON JavaScript Object Notation (JSON) is a file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types.

KB In DLs, a Knowledge Base (KB) is the combination of a TBox and an ABox [18]. It is used to represent domain knowledge in an explicit and structured way that facilitates its application in knowledge based systems.

KDD Knowledge Discovery in Databases (KDD) is the non- trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [63].

KPI KPIs are defined as quantifiable and strategic measurements that reflects an organisation's critical success factors. Key Performance Indicators are very important for understanding and improving manufacturing performance; both from the lean manufacturing perspective of eliminating waste and from the corporate perspective of achieving strategic goals [90].

LTL In logic, linear temporal logic or linear-time temporal logic (LTL) is a modal temporal logic with modalities referring to time [143].

MES Manufacturing Execution Systems (MES) offer a meaningful functional addition to plan and control direct all manufacturing processes, to ensure process transparency, and to map the flow of material to information within the supply chain [173].

MTL Metric Temporal Logic (MTL) is a logic designed by Koymans [111] and can be used to reason over temporal systems.

NC Numerical Control (NC) is the automated control of machining tools (drills, boring tools, lathes) and 3D printers by means of a computer.

OBDA In OBDA, a conceptual layer is provided in the form of an ontology that defines a shared vocabulary, models the domain, hides the structure of the data sources, and enriches incomplete data with background knowledge. Then, queries are posed over this high-level conceptual view, and the users no longer need an understanding of the data sources, the relation between them, or the encoding of the data [41].

OBDI Ontology- Based Data Integration (OBDI) refers to the use of (potentially several layers of) ontologies that capture implicit knowledge across heterogeneous data sources to achieve semantic interoperability between these sources.

OBSSDI Ontology-Based Stream-Static Data Integration (OBSSDI), a data integration approach for static and streaming data as proposed by [102].

OWL The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring Ontologies [140].

PLR Part-load ratio (PLR) is the ratio between actual load of a given machine and its available capacity.

PoC A proof of concept refers to evidence, typically deriving from an experiment or pilot project, which demonstrates that a design concept, business proposal, etc. is feasible.

RDBMS A relatiaonal database management system (RDBMS) is a DBMS based on the relational model. Most databases in widespread use today are based on his relational database model.

RDF The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modelling of information that is implemented in web resources, using a variety of syntax notations and data serialisation formats. It is also used in knowledge management applications [114].

RDFS Resource Description Framework Schema (RDFS) is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources. RDF Schema is written in RDF [75].

SCADA Supervisory control and data acquisition (SCADA) is a control system architecture that uses computers, networked data communications and graphical user interfaces for high-level process supervisory management..

SOSA Sensor, Observation, Sample, and Actuator (SOSA) Ontology, a subset of SSN.

SPARQL SPARQL (SPARQL Protocol and RDF Query Language) is an RDF query language able to retrieve and manipulate data stored in RDF format [157].

SQL Structured Query Language (SQL) is a domain-specific language used in programming and designed for managing data held in a RDBMS.

SSN Semantic Sensor Network Ontology (SSN)[55].

TQL Temporal Query Language (TQL), an interval-based query language [78].

TRL Technology Readiness Level (TRL), is a way to measure and classify the technological maturity of a given technology [85].

URI A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource.

VAD Vibration Assisted Drilling (VAD) referes to drilling processes that are combined with axial vibration.

W3C World Wide Web Consortium (W3C) is the main international standards organisation for the World Wide Web (abbreviated WWW or W3). Founded and currently led by Tim Berners-Lee, the consortium is made up of member organisations which maintain full-time staff to work together in the development of standards for the World Wide Web [166].

XML Extensible Markup Language (XML), describes a class of data objects called XML documents and partially describes the behaviour of computer programs which process them. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language [88]. By construction, XML documents are conforming SGML documents.XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure[166].

Bibliography

[1] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of databases*. Addison-Wesley, 1995.

[2] Serge Abiteboul, Marcelo Arenas, Pablo Barceló, Meghyn Bienvenu, Diego Calvanese, Claire David, Richard Hull, Eyke Hüllermeier, Benny Kimelfeld, Leonid Libkin, Wim Martens, Tova Milo, Filip Murlak, Frank Neven, Magdalena Ortiz, Thomas Schwentick, Julia Stoyanovich, Jianwen Su, Dan Suciu, Victor Vianu, and Ke Yi. Research Directions for Principles of Data Management. *ACM SIGMOD Record*, 45(4):5–17, may 2017.

[3] Andrea Acciarri, Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Mattia Palmieri, and Riccardo Rosati. Quonto: Querying Ontologies. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1670–1671, 2005.

[4] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient Similarity Search in Sequence Databases. In *Foundations of Data Organization and Algorithms, 4th International Conference, FODO'93, Chicago, Illinois, USA, October 13-15, 1993, Proceedings*, pages 69–84, 1993.

[5] James Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[6] Henrik André-Jönsson and Dushan Z Badal. Using Signature Files for Querying Time-Series Data. In *European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD)*, volume 1263, pages 211–220, 1997.

[7] Álvar Arnaiz-González, Asier Fernández-Valdivielso, Andres Bustillo, and Luis Norberto López de Lacalle. Using artificial neural networks for the prediction of dimensional error on inclined surfaces manufactured by ball-end milling. *International Journal of Advanced Manufacturing Technology*, 83(5-8):847–859, 2016.

[8] Alessandro Artale, Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev. Temporal Description Logic for Ontology-Based Data Access. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 711–717, 2013.

[9] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. First-order rewritability of temporal ontology-mediated queries. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2706–2712, 2015.

[10] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. Tractable interval temporal propositional and description logics. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1417–1423, 2015.

[11] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Ontology-Mediated Query Answering over Temporal Data: A Survey. In *24th International Symposium on Temporal Representation and Reasoning, TIME 2017, October 16-18, 2017, Mons, Belgium*, pages 1:1—-1:37, 2017.

[12] Franz Baader and Ulrike Sattler. Description logics with aggregates and concrete domains. *Inf. Syst.*, 28(8):979–1004, 2003.

[13] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 364–369, 2005.

[14] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3):21:1—-21:32, 2012.

[15] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporalizing ontology-based data access. In M.P. Bonacina, editor, *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, pages 330–344. Springer, Berlin, Heidelberg, 2013.

[16] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal query entailment in the Description Logic SHQ. *Journal of Web Semantics*, 33:71–93, 2015.

[17] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal conjunctive queries in expressive description logics with transitive roles. In *AI 2015: Advances in Artificial Intelligence - 28th Australasian Joint Conference, Canberra, ACT, Australia, November 30 - December 4, 2015, Proceedings*, volume 9457, pages 21–33. Springer, Cham, nov 2015.

[18] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[19] V. Baghlani, P. Mehbudi, J. Akbari, and M. Sohrabi. Ultrasonic Assisted Deep Drilling of Inconel 738LC Superalloy. *Procedia CIRP*, 6:571–576, jan 2013.

[20] Jon Barwise. An Introduction to First-Order Logic. *Studies in Logic and the Foundations of Mathematics*, 90:5–46, 1977.

[21] Alessandro Beghi, Luca Cecchinato, and Mirco Rampazzo. A multi-phase genetic algorithm for the efficient management of multi-chiller systems. *Energy Conversion and Management*, 52(3):1650–1661, 2011.

[22] Michael Benedikt and Michael. How can reasoners simplify database querying (and why haven't they done it yet)? In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems - SIGMOD/PODS '18*, volume 15, pages 1–15, New York, New York, USA, 2018. ACM Press.

[23] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, May:28–37, 2001.

[24] Stefan Biffl, Arndt Lüder, and Dietmar Winkler. Multi-Disciplinary Engineering for Industrie 4.0: Semantic Challenges and Needs. In *Semantic Web Technologies for Intelligent Engineering Applications*, pages 17–51. Springer International Publishing, Cham, 2016.

[25] Friedrich Bleicher, Benjamin Mörzinger, Christoph Loschan, Ines Leobner, Peter Smolek, Wolfgang Kastner, Bernhard Heinzl, Iva Kovacic, Georgios Gourlis, Thomas Sobottka, Felix Kamhuber, Roman Klug, Philipp Trummer, Michael Mühlhauser, Neat Likaj, Gerhard Gotz, Klemens Gregor Schulmeister, Günther Daubner, Rudolf Zeman, Niki Popper, Matthias Rössler, Thomas Palatin, Erich Krall, Josef Obiltschnig, Karsten Buchholz, Ewald Perwög, Friedrich Koidl, Gerhard Fritz, and Albert Dechant. Balanced Manufacturing (BaMa). Technical report, TU Wien- Institute for Production Engineering and Laser Technology, Vienna, 2018.

[26] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporal Query Answering in the Description Logic DL-Lite. In P. Fontaine, C. Ringeissen, and R.A. Schmidt, editors, *Frontiers of Combining Systems. FroCoS 2013. Lecture Notes in Computer Science*, pages 165–180. Springer, Berlin, Heidelberg, 2013.

[27] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporalizing rewritable query languages over knowledge bases. *Journal of Web Semantics*, 33:50–70, aug 2015.

[28] Sebastian Brandt, Elem Güzel Kalayci, Roman Kontchakov, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Ontology-based data access with a horn fragment of metric temporal logic. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1070–1076, 2017.

[29] Sebastian Brandt, Elem Güzel Kalayci, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. A framework for temporal ontology-based data access: A proposal. In *New*

Trends in Databases and Information Systems - ADBIS 2017 Short Papers and Workshops, AMSD, BigNovelTI, DAS, SW4CH, DC, Nicosia, Cyprus, September 24-27, 2017, Proceedings, pages 161–173, 2017.

[30] Sebastian Brandt, Diego Calvanese, Elem Güzel Kalayci, Roman Kontchakov, Benjamin Mörzinger, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Two-dimensional rule language for querying sensor log data: a framework and use cases. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2013, Proceedings, Part II, submitted*, 2019.

[31] Andres Bustillo, José Francisco Díez-Pastor, Guillem Quintana, and César García-Osorio. Avoiding neural network fine tuning by using ensemble learning: Application to ball-end milling operations. *International Journal of Advanced Manufacturing Technology*, 57(5-8): 521–532, 2011.

[32] Andres Bustillo, Maritza Correa, A Bustillo, and M Correa. Using artificial intelligence to predict surface roughness in deep drilling of steel components. *Journal for Intelligent Manufacturing*, 23:1893–1902, 2012.

[33] Andres Bustillo, Daniil Yu Pimenov, Maciej Matuszewski, and Tadeusz Mikolajczyk. Using artificial intelligence models for the prediction of surface wear based on surface isotropy levels. *Robotics and Computer-Integrated Manufacturing*, 53:215–227, oct 2018.

[34] Andrea Calì, Georg Gottlob, and Andreas Pieris. Advanced processing for ontological queries. *Proceedings of the VLDB Endowment*, 3(1):554–565, 2010.

[35] Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *Artificial Intelligence*, 193:87–128, 2012.

[36] D. Calvanese, E. Kharlamov, W. Nutt, and C. Thorne. Aggregate queries over ontologies. In *Proc. of the 2nd Int. Workshop on Ontologies and Information Systems for the Semantic Web, ONISW 2008*, pages 97–104, 2008.

[37] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[38] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The Mastro System for Ontology-based Data Access. *Semantic Web*, pages 1–11, 2011.

[39] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Marco Montali, and Ario Santoso. Semantically-governed data-aware processes. In *CEUR Workshop Proceedings*, volume 861, pages 21–32, 2012.

[40] Diego Calvanese, Elem Güzel Kalayci, Vladislav Ryzhikov, and Guohui Xiao. Towards practical OBDA with temporal ontologies (position paper). *Lecture Notes in Computer Science*, 9898:18–24, 2016.

[41] Diego Calvanese, Cogrel Benjamin, Sarah Komla-Ebri, , Roman Kontchakov, Davide Lanti, Martin Rezk, Rodriguez-Muro Xiao, and Guohui. Ontop : Answering SPARQL Queries over Relational Databases. *Semantic Web Journal*, 8(3):471–487, 2017.

[42] Diego Calvanese, Tahir Emre Kalayci, Marco Montali, and Stefano Tinella. Ontology-based data access for extracting event logs from legacy data: The onprom tool and methodology. In *Lecture Notes in Business Information Processing*, volume 288, pages 220–236. Springer, Cham, 2017.

[43] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. Lodlive, exploring the web of data. In *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*, pages 197–200, 2012.

[44] Kaushik Chakrabarti, Eamonn Keogh, Sharad Mehrotra, and Michael Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Transactions on Database Systems*, 27(2):188–228, jun 2002.

[45] Peter Pin-Shan Chen. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[46] Chen Fu Chien and Shih Chung Chuang. A framework for root cause detection of sub-batch processing system for semiconductor manufacturing big data analytics. *IEEE Transactions on Semiconductor Manufacturing*, 27(4):475–488, 2014.

[47] Chen Fu Chien, Chia Yu Hsu, and Pei Nong Chen. Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence. *Flexible Services and Manufacturing Journal*, 25(3):367–388, 2013.

[48] Chen Fu Chien, Shao Chung Hsu, and Ying Jen Chen. A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence. *International Journal of Production Research*, 51(8):2324–2338, 2013.

[49] Chen-Fu Chien, Kuo-Hao Chang, Wen-Chih Wang, C.-F Chien, K.-H Chang, and W.-C Wang. An empirical study of design-of-experiment data mining for yield-loss diagnosis for semiconductor manufacturing. *Journal for Intelligent Manufacturing*, 25:961–972, 2014.

[50] Chen Fu Chien, Chiao Wen Liu, and Shih Chung Chuang. Analysing semiconductor manufacturing big data for root cause detection of excursion for yield enhancement. *International Journal of Production Research*, 55(17):5095–5107, 2015.

[51] Jongsawas Chongwatpol. Prognostic analysis of defects in manufacturing. *Industrial Management & Data Systems*, 115(1):64–87, 2015.

[52] Alok Kumar Choudhary, Jenny A. Harding, and Manoj Kumar Tiwari. Data mining in manufacturing: a review based on the kind of knowledge. *J. Intelligent Manufacturing*, 20(5): 501–521, 2009.

[53] Kian Jon Chua, Siaw Kiang Chou, Wenming Yang, and Jinyue Yan. Achieving better energy-efficient air conditioning: A review of technologies and strategies. *Applied Energy*, 104:87–104, 2013.

[54] Cristina Civili, Marco Console, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Lorenzo Lepore, Riccardo Mancini, Antonella Poggi, Riccardo Rosati, Marco Ruzzi, Valerio Santarelli, and Domenico Fabio Savo. MASTRO STUDIO: Managing Ontology-Based Data Access applications. In *Proceedings of the VLDB Endowment 6*, number 12, pages 1314–1317. Springer, Cham, 2013.

[55] Simon Cox, Kerry Taylor, Krzysztof Janowicz, Armin Haller, Maxime Lefrançois, and Danh Le Phuoc. Semantic sensor network ontology. W3C recommendation, W3C, 2017.

[56] Jim Crompton. Putting the FOCUS on Data. In *W3C Workshop on Semantic Web in Oil & Gas Industry*, 2008.

[57] Walter Dean. Computational complexity theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition, 2016.

[58] Berend Denkena, Justin Schmidt, and Max Krüger. Data Mining Approach for Knowledge-based Process Planning. *Procedia Technology*, 15:406–415, 2014.

[59] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik Van De Walle. RML: A generic language for integrated RDF mappings of heterogeneous data. In *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW) 2014, Seoul, Korea, April 8, 2014Data mining and knowledge discovery in databases*, volume 1184, 2014.

[60] Xin Luna Dong, Anastasios Kementsietsidis, and Wang-Chiew Tan. A Time Machine for Information: Looking Back to Look Forward. In *Proceedings of the VLDB Endowment*, volume 45, pages 23–32, 2016.

[61] Fajar J Ekaputra, Marta Sabou, Estefanía Serral, Elmar Kiesling, and Stefan Biffl. Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review. *Open Journal of Information Systems (OJIS)*, 4(1), 2017.

[62] Philippe Esling and Carlos Agon. Time-Series Data Mining. *Advanced Information and Knowledge Processing*, 45(1):12:1—-12:34, 2012.

[63] Usama M. Fayyad and Ramasamy Uthurusamy. Data mining and knowledge discovery in databases. *Communications of the ACM*, 39(11):24–26, 1996.

[64] Stefan Feldmann, Sebastian J.I. Herzig, Konstantin Kernschmidt, Thomas Wolfenstetter, Daniel Kammerl, Ahsan Qamar, Udo Lindemann, Helmut Krcmar, Christiaan J.J. Paredis, and Birgit Vogel-Heuser. Towards effective management of inconsistencies in model-based engineering of automated production systems. *IFAC-PapersOnLine*, 48(3):916–923, jan 2015.

[65] Florian Kloibhofer. Simulationsbasierte Optimierung der Betriebsstrategie von energietechnischen Anlagen. Master's thesis, TU Wien, 2018.

[66] Tak Chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, feb 2011.

[67] A Garg, K Tai, V Vijayaraghavan, and Pravin M Singru. Mathematical modelling of burr height of the drilling process using a statistical-based multi-gene genetic programming approach. *International Journal of Advanced Manufacturing Technology*, 73(1-4):113–126, 2014.

[68] Saurabh Garg, Surjya K. Pal, and Debabrata Chakraborty. Evaluation of the performance of backpropagation and radial basis function neural networks in predicting the drill flank wear. *Neural Computing and Applications*, 16(4-5):407–417, 2007.

[69] Martin Giese, Ahmet Soylu, Guillermo Vega-Gorgojo, Arild Waaler, Peter Haase, Ernesto Jiménez-Ruiz, Davide Lanti, Martin Rezk, Guohui Xiao, Özgür Özçep, and Riccardo Rosati. Optique: Zooming in on Big Data. *IEEE Computer*, 48(3):60–67, 2015.

[70] Dejan Gradišar, Miha Glavan, Stank Strmčnik, and Gašper Mušič. ProOpter: An advanced platform for production analysis and optimization. *Computers in Industry*, 70:102–115, 2015.

[71] Irlán Grángel-Gonzalez, Paul Baptista, Lavdim Halilaj, Steffen Lohmann, Maria Esther Vidal, Christian Mader, and Sören Auer. The industry 4.0 standards landscape from a semantic integration perspective. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, pages 1–8, 2018.

[72] Markus Graube, Leon Urbas, and Jan Hladik. Integrating industrial middleware in Linked Data collaboration networks. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, volume 2016-Novem, pages 1–8. IEEE, sep 2016.

[73] Karl-Heinrich Grote and Jörg Feldhusen, editors. *Dubbel*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[74] Maciej Grzenda, Andres Bustillo, Pawel Zawistowski, Maciej Grzenda, Andreas Bustillo, and Pawel Zawistowski. A soft computing system using intelligent imputation strategies for roughness prediction in deep drilling. *Journal for Intelligent Manufacturing*, 23:1733–1743, 2012.

[75] Ramanathan Guha and Dan Brickley. RDF Schema 1.1. Technical report, W3C, 2014.

[76] Amit Kumar Gupta, Sharath Chandra Guntuku, Raghuram Karthik Desu, and Aditya Balu. Optimisation of turning parameters by integrating genetic algorithm with support vector regression and artificial neural networks. *International Journal of Advanced Manufacturing Technology*, 77(1-4):331–339, 2015.

[77] Munish Kumar Gupta, P K Sood, and Vishal S Sharma. Machining Parameters Optimization of Titanium Alloy using Response Surface Methodology and Particle Swarm Optimization under Minimum-Quantity Lubrication Environment. *Materials and Manufacturing Processes*, 31(13):1671–1682, 2016.

[78] Víctor Gutiérrez-Basulto and Szymon Klarman. Towards a Unifying Approach to Representing and Querying Temporal Data in Description Logics. In M. Krötzsch and U. Straccia, editors, *Web Reasoning and Rule Systems. RR 2012. Lecture Notes in Computer Science*, pages 90–105. Springer, Berlin, Heidelberg, 2012.

[79] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Roman Kontchakov. Temporalized EL ontologies for accessing temporal data: Complexity of atomic queries. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1102–1108, 2016.

[80] Peter Haase, Ian Horrocks, Dag Hovland, Thomas Hubauer, Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Johan W. Klüwer, Christoph Pinkel, Riccardo Rosati, Valerio Santarelli, Ahmet Soylu, and Dmitriy Zheleznyakov. Optique system: towards ontology and mapping management in OBDA solutions. In *Proceedings of the Second International Workshop on Debugging Ontologies and Ontology Mappings, Montpellier, France, May 27, 2013*, pages 21–32, 2013.

[81] Matthias Hacksteiner, Fabian Duer, Daniel Finkeldei, Martin Obermair, and Friedrich Bleicher. Monitoring of process power and energy during machining. In *International Conference on High Speed Machining*, 2016.

[82] Joseph Y. Halpern. A Propositional Modal Logic of Time Intervals. *ACM*, 38(4):935–962, 1991.

[83] Jenny A. Harding, M Shahbaz, and A Kusiak. Data Mining in Manufacturing: A Review. *Journal of Manufacturing Science and Engineering*, 128:969–976, 2006.

[84] Babak Bagheri Hariri, Diego Calvanese, Marco Montali, Ario Santoso, and Dmitry Solomakhin. Verification of semantically-enhanced artifact systems. In *Service-Oriented Computing - 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings*, pages 600–607, 2013.

[85] Mihály Héder. From NASA to EU: The evolution of the TRL scale in Public Sector Innovation. *Innovation Journal*, 22(2):1–23, 2017.

[86] Bernhard Heinzl, Irene Hafner, Peter Smolek, Ines Leobner, Georgios Gourlis, Martin Obermair, Nikolas Popper, and Wolfgang Kastner. Towards a Common Description of Interdisciplinary Aspects Relevant for Holistic Energy Analysis of Production Facilities. In *Workshop der ASIM/GI-Fachgruppen STS/GMMS*, 2015.

[87] Ian Horrocks, Thomas Hubauer, Ernesto Jimenez-Ruiz, Evgeny Kharlamov, Manolis Koubarakis, Ralf Möller, Konstantina Bereta, Christian Neuenstadt, Özgür Özçep, Mikhail Roshchin, Panayiotis Smeros, and Dmitriy Zheleznyakov. Addressing streaming and historical data in OBDA systems: Optique's approach (statement of interest). In *CEUR Workshop Proceedings*, volume 992, pages 33–40, 2013.

[88] ISO. ISO 8879- information processing – text and office systems – standard generalized markup language. Iso, International Organization for Standardization, Geneva, Switzerland, 1986.

[89] ISO. ISO 14644- cleanrooms and associated controlled environments- part 1: Classification of air cleanliness. Iso, International Organization for Standardization, Geneva, Switzerland, 1999.

[90] ISO. ISO 22400- automation systems and integration - key performance indicators (KPIs) for manufacturing operations management - part 2: Definitions and descriptions. Iso, International Organization for Standardization, Geneva, Switzerland, 2014.

[91] Jérémy Jallageas, Jean-Yves K'nevez, Mehdi Chérif, and Olivier Cahuc. Modeling and optimization of vibration-assisted drilling on positive feed drilling unit. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1205–1216, 2013.

[92] Elem Güzel Kalaycı. *Ontology- based access to temporal data*. PhD thesis, Free University Bolzano, 2019.

[93] Elem Güzel Kalaycı and Diego Calvanese. Ontop-temporal : A Tool for Ontology-based Query Answering over Temporal Data. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1927–1930, Torino, Italy, 2018. ACM.

[94] Bernard Kamsu-Foguem, Fabien Rigal, and Félix Mauget. Mining association rules for the quality improvement of the production process. *Expert Systems with Applications*, 40(4): 1034–1045, 2013.

[95] Seokho Kang, Eunji Kim, Jaewoong Shim, Wonsang Chang, and Sungzoon Cho. Product failure prediction with missing data. *International Journal of Production Research*, 56:4849–4859, 2017.

[96] Seokho Kang, Eunji Kim, Jaewoong Shim, Sungzoon Cho, Wonsang Chang, and Junhwan Kim. Mining the relationship between production and customer service data for failure analysis of industrial products. *Computers & Industrial Engineering*, 106:137–146, 2017.

[97] Girish Kant and Kuldip Singh Sangwan. Predictive modeling for power consumption in machining using artificial intelligence techniques. In *Procedia CIRP*, volume 26, pages 403–407, 2015.

[98] Natalya Keberle. Answering conjunctive queries over a temporally-ordered finite sequence of aboxes sharing one tbox. In *Proceedings of the 9th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer, Kherson, Ukraine, June 19-22, 2013*, pages 79–90, 2013.

[99] Eamonn Keogh. A fast and robust method for pattern matching in time series databases. In *Proceedings of 9th International Conference on Tools*, pages 145–150, 1997.

[100] Evgeny Kharlamov, Nina Solomakhina, Özgür Lütfü Özçep, Dmitriy Zheleznyakov, Thomas Hubauer, Steffen Lamparter, Mikhail Roshchin, Ahmet Soylu, and Stuart Watson. How Semantic Technologies Can Enhance Data Access at Siemens Energy. In *International Semantic Web Conference*, pages 601–619, Cham, 2014. Springer International Publishing.

[101] Evgeny Kharlamov, Yannis Kotidis, Theofilos Mailis, Christian Neuenstadt, Charalampos Nikolaou, Özgür Özçep, Christoforos Svingos, Dmitriy Zheleznyakov, Sebastian Brandt, Ian Horrocks, Yannis Ioannidis, Steffen Lamparter, and Ralf Möller. Towards analytics aware ontology based access to static and streaming data. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, volume 9982 LNCS, pages 344–362, 2016.

[102] Evgeny Kharlamov, Christoforos Svingos, Dmitriy Zheleznyakov, Ian Horrocks, Yannis Ioannidis, Ralf Moeller, Sebastian Brandt, Ernesto Jimenez-Ruiz, Yannis Kotidis, Steffen Lamparter, Theofilos Mailis, Christian Neuenstadt, Öezguer Öezçep, and Christoph Pinkel. Ontology-Based Integration of Streaming and Static Relational Data with Optique. In *Proceedings of the 2016 International Conference on Management of Data - SIGMOD '16*, pages 2109–2112, New York, New York, USA, 2016. ACM Press.

[103] Evgeny Kharlamov, Dag Hovland, Martin G. Skjæveland, Dimitris Bilidas, Ernesto Jiménez-Ruiz, Guohui Xiao, Ahmet Soylu, Davide Lanti, Martin Rezk, Dmitriy Zheleznyakov, Martin Giese, Hallstein Lie, Yannis Ioannidis, Yannis Kotidis, Manolis Koubarakis, and Arild Waaler. Ontology Based Data Access in Statoil. *Journal of Web Semantics*, 44:3–36, 2017.

[104] Evgeny Kharlamov, Theofilos Mailis, Gulnar Mehdi, Christian Neuenstadt, Özgür Özçep, Mikhail Roshchin, Nina Solomakhina, Ahmet Soylu, Christoforos Svingos, Sebastian Brandt, Martin Giese, Yannis Ioannidis, Steffen Lamparter, Ralf Möller, Yannis Kotidis, and Arild Waaler. Semantic access to streaming and static data at Siemens. *Journal of Web Semantics*, 44:54–74, 2017.

[105] Szymon Klarman and Thomas Meyer. Querying Temporal Databases via OWL 2 QL. In *Web Reasoning and Rule Systems. RR 2014. Lecture Notes in Computer Science*, pages 92–107. Springer, Cham, 2014.

[106] Gülser Köksal and Inci Batmaz. A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems With Applications*, 38:13448–13467, 2011.

[107] Roman Kontchakov, Carsten Lutz, David Toman, Franz Wolter, and Michael Zakharyaschev. The Combined Approach to Query Answering in DL-Lite. *Artificial Intelligence*, pages 247–257, 2010.

[108] Roman Kontchakov, Laura Pandolfo, Luca Pulina, Vladislav Ryzhikov, and Michael Zakharyaschev. Temporal and Spatial OBDA with Many-Dimensional Halpern-Shoham Logic. In *Proceed- ings of 25th International Joint Conference on Artificial Intelligence (IJCAI-2016).*, pages 1160–1166. AAAI Press, 2016.

[109] E. V. Kostylev and J. L. Reutter. Complexity of answering counting aggregate queries over DL-Lite. *J. Web Semantics*, 33:94–111, 2015.

[110] Olga Kovalenko, Estefanía Serral, Marta Sabou, Fajar J. Ekaputra, Dietmar Winkler, and Stefan Biffl. Automating Cross-Disciplinary Defect Detection in Multi-disciplinary Engineering Environments. In *Knowledge Engineering and Knowledge Management - 19th International Conference, EKAW 2014, Linköping, Sweden, November 24-28, 2014. Proceedings*, pages 238–249. Springer, Cham, 2014.

[111] Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[112] Andrew Kusiak. Data mining: Manufacturing and service applications. *International Journal of Production Research*, 44:4175–4191, sep 2006.

[113] Andrew Kusiak. Smart manufacturing. *International Journal of Production Research*, 56(1-2): 508–517, jan 2018.

[114] Ora Lassila. Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C, 1999.

[115] Marc Lavielle. Using penalized contrasts for the change-point problem. *Signal Processing*, 85 (8):1501–1510, 2005.

[116] Freddy Lécué and Jeff Z Pan. Predicting knowledge in an ontology stream. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 2662–2669, 2013.

[117] Sseverin Lemaignan, Ali Siadat, Jean-Yves Dantan, and Anatoli Semenenko. MASON: A Proposal For An Ontology Of Manufacturing Domain. In *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06)*, pages 195–200. IEEE, 2006.

[118] Ines Leobner, Peter Smolek, Bernhard Heinzl, Iva Kovacic, Karl Ponweiser, Walter Mayrhofer, Wolfgang Kastner, and Fabian Dür. Balanced Manufacturing – a Methodology for Energy Efficient Production Plant Operation. In *Proceedings of the 10th Conference on Sustainable Development of Energy, SDEWES2015.0268*, pages 1–11, 2015.

[119] Xiaonan Li and Sigurdur Olafsson. Discovering Dispatching Rules Using Data Mining. *Journal of Scheduling*, 8(6):515–527, 2005.

[120] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[121] Ibrahem Maher, M. E.H. Eltaib, Ahmed A.D. Sarhan, and R M El-Zahry. Cutting force-based adaptive neuro-fuzzy approach for accurate surface roughness prediction in end milling operation for intelligent machining. *International Journal of Advanced Manufacturing Technology*, 76(5-8):1459–1467, 2014.

[122] Theophano. Mitsa. *Temporal Data Mining*. Chapman and Hall/CRC, New York, 2010.

[123] Ralf Möller, Christian Neuenstadt, and Özgür L Özçep. Stream-temporal Querying with Ontologies. In *Proceedings of the 1st Workshop on High-Level Declarative Stream Processing co-located with the 38th German AI conference (KI 2015), Dresden, Germany, September 22, 2015.*, pages 42–55, 2015.

[124] Danielle Monfet and Radu Zmeureanu. Identification of the electric chiller model for the energyplus program using monitored data in an existing cooling plant. In *Proceedings of Building Simulation 2011*, pages 530–537, 2011.

[125] Danielle Monfet and Radu Zmeureanu. Calibration of a central cooling plant model using manufacturers data and measured input parameters and comparison with measured performance. In *Proceedings of BS2013: 13th Conference of International Building Performance Simulation Associati*, page 8, Chambéry, France, 2012.

[126] Danielle Monfet and Radu Zmeureanu. Calibration of an energyplus central cooling plant model with masurements and inter-program comparison. In *Proceedings of BS2013*, pages 622–629, 2013.

[127] László Monostori. Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. *Procedia CIRP*, 17:9–13, 2014.

[128] Benjamin Mörzinger, Marta Sabou, Fajar J Ekaputra, and Nikolaus Sindelar. Improving industrial optimization with Semantic Web technologies. In Maria Koutraki Ali Khalili, editor, *Proceedings of the Posters and Demos Track of the 14th International Conference on Semantic Systems*, page 4, Vienna, 2018.

[129] Benjamin Mörzinger, Christoph Loschan, Florian Kloibhofer, and Friedrich Bleicher. A modular, holistic optimization approach for industrial appliances. In *Procedia CIRP*, volume 79, pages 551–556. Elsevier, jan 2019.

[130] Bogdan Nedelcu. About Big Data and its Challenges and Benefits in Manufacturing. Technical Report 3, 2013.

[131] Christian Neuenstadt, Ralf Möller, and Özgür L Özçep. OBDA for Temporal Querying and Streams. In *HiDeSt@KI 2015*, pages 70–75, 2015.

[132] Petr Novák and Radek Šindelář. Applications of Ontologies for Assembling Simulation Models of Industrial Systems. In *On the Move to Meaningful Internet Systems: OTM 2011 Workshops - Confederated International Workshops and Posters: EI2N+NSF ICE, ICSP+INBAST, ISDE, ORM, OTMA, SWWS+MONET+SeDeS, and VADER 2011, Hersonissos, Crete, Greece, October 17-21, 2011. Proceedings*, volume 7046 LNCS, pages 148–157, 2011.

[133] Petr Novak and Radek Sindelar. Semantic design and integration of simulation models in the industrial automation area. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, pages 1–8. IEEE, 2012.

[134] DOE (Department of Energy). DOE 2 Reference manual, part 1, version 2.1. Technical report, DOE, Lawrence Berkeley National Laboratories, Berkeley, Calif, 1980.

[135] Kiyoshi Okamura, Hiroyuki Sasahara, Toshiaki Segawa, and Masaomi Tsutsumi. Low-Frequency Vibration Drilling of Titanium Alloy. *JSME International Journal Series*, 1(46): 76–82, 2006.

[136] Özgür L. Özçep, Möller Ralf, Christian Neuenstadt, Dmitriy Zheleznyakov, and Evgeny Kharlamov. A Semantics for Temporal and Stream-Based Query Answering in an OBDA Context. Technical Report November 2012, Hamburg University of Technology, Hamburg, 2013.

[137] Özgür Lütfü Özçep, Ralf Möller, and Christian Neuenstadt. A stream-temporal query language for ontology based data access. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.*, pages 696–708, 2014.

[138] Atakan Öztürk, Sinan Kayalıgil, and Nur E Zdemirel. Manufacturing lead time estimation using data mining. *European Journal of Operational Research*, 173:683–700, 2005.

[139] Sudhansu Panda, Ankita K. Singh, Debabrata Chakraborty, and Surjya Pal. Drill wear monitoring using back propagation neural network. *Journal of Materials Processing Technology*, 172 (2):283–290, feb 2006.

[140] Peter Patel-Schneider, Sebastian Rudolph, Bijan Parsia, Markus Krötzsch, and Pascal Hitzler. OWL 2 Web Ontology Language Primer (Second Edition). Technical report, W3C, 2012.

[141] Oliver Pecat. *Vibrationsunterstütztes Bohren von Werkstoffverbunden aus CFK und Titan*. PhD thesis, Universität Bremen, 2018.

[142] Niklas Petersen, Lavdim Halilaj, Irlán Grangel-González, Steffen Lohmann, Christoph Lange, and Sören Auer. Realizing an RDF-based information model for, a manufacturing company – A case study. In *The Semantic Web – ISWC 2017*, pages 350–366. Springer International Publishing, 2017.

[143] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57. [Institute of Electrical and Electronics Engineers], 1977.

[144] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking Data to Ontologies. *Journal on Data Semantics*, 10: 133–173, 2008.

[145] Yunyao Qu, C Wang, and XS Wang. Supporting fast search in time series for movement patterns in multiple scales. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 251–258, New York, New York, USA, 1998. ACM Press.

[146] Tobias Rackow, Tallal Javied, Toni Donhauser, C Martin, Peter Schuderer, and Jörg Franke. Green Cockpit: Transparency on Energy Consumption in Manufacturing Companies. In *Proceedings of 12th Conference on Sustainable Development of Energy, Water and Environment Systems*, volume 26, pages 498–503, 2015.

[147] Yves Raimond and Guus Schreiber. RDF 1.1 primer. Technical report, W3C, 2014.

[148] Luis Ramos. Semantic Web for manufacturing, trends and open issues: Toward a state of the art. *Computers and Industrial Engineering*, 90:444–460, 2015.

[149] Sanjay Rawat and Helmi Attia. Characterization of the dry high speed drilling process of woven composites using Machinability Maps approach. *CIRP Annals*, 58(1):105–108, 2009.

[150] Christina Reuter and Felix Brambring. Improving Data Consistency in Production Control. *Procedia CIRP*, 41:51–56, 2016.

[151] Christina Reuter, Felix Brambring, Jan Weirich, and Arne Kleines. Improving Data Consistency in Production Control by Adaptation of Data Mining Algorithms. In *Procedia CIRP*, volume 56, pages 545–550, 2016.

[152] Juan J Rodríguez, Guillem Quintana, Andres Bustillo, and Joaquim Ciurana. A decision-making tool based on decision trees for roughness prediction in face milling. *International Journal of Computer Integrated Manufacturing*, 30(9):943–957, 2017.

[153] Thomas A. Runkler. Datenvorverarbeitung. In *Data Mining*, pages 21–34. Vieweg+Teubner, Wiesbaden, 2010. doi: 10.1007/978-3-8348-9353-6_3.

[154] Guenther Schuh, Till Potente, Christina Thomas, and Felix Brambring. Improving Data Integrity in Production Control. In *Procedia CIRP*, volume 9, pages 44–48. Elsevier, jan 2013.

[155] Günther Schuh, Christina Thomas, Annika Hauptvogel, and Felix Brambring. Achieving Higher Scheduling Accuracy in Production Control by Implementing Integrity Rules for Production Feedback Data. In *Procedia CIRP*, volume 19, pages 142–147. Elsevier, jan 2014.

[156] Günther Schuh, Christina Reuter, Jan Philipp Prote, Felix Brambring, and Julian Ays. Increasing data integrity for improving decision making in production planning and control. *CIRP Annals - Manufacturing Technology*, 66(1):425–428, 2017.

[157] Andy Seaborne and Steven Harris. SPARQL 1.1 Query Language. Technical report, W3C, 2013.

[158] M Shahbaz, Srinivas, Jenny Harding, and M Turner. Product design and manufacturing process improvement using association rules. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 220(2):243–254, feb 2006.

[159] Hagit Shatkay and S.B. Zdonik. Approximate queries and representations for large data sequences. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 536–545. IEEE Comput. Soc. Press, 1996.

[160] Seung-Jun Shin, Jungyub Woo, and Sudarsan Rachuri. Predictive analytics model for power consumption in manufacturing. *Procedia CIRP*, 15:153–158, 2014.

[161] Rupinder Singh and J. S. Khamba. Ultrasonic machining of titanium and its alloys: A review. *Journal of Materials Processing Technology*, 173(2):125–135, 2006.

[162] Thomas Sobottka, Felix Kamhuber, and Wilfried Sihn. Increasing energy efficiency in production environments through an optimized, hybrid simulation-based planning of production and its periphery. In *Proceedings of the 24th CIRP Conference on Life Cycle Engineering*, pages 440–445, Kamakura, Japan, 2017.

[163] Ahmet Soylu, Martin Giese, Ernesto Jimenez-Ruiz, Evgeny Kharlamov, Rudolf Schlatte, Christian Neuenstadt, Özgür Özçep, Hallstein Lie, Vidar Klungre, Sebastian Brandt, and Ian Horrocks. Ontology-based visual querying with OptiqueVQS: Statoil and Siemens cases. In *CEUR Workshop Proceedings*, volume 1818, pages 34–39, 2016.

[164] Ahmet Soylu, Evgeny Kharlamov, Dmitriy Zheleznyakov, Ernesto Jimenez-Ruiz, Martin Giese, Martin G Skjaeveland, Dag Hovland, Rudolf Schlatte, Sebastian Brandt, Hallstein Lie, and Ian Horrocks. OptiqueVQS: a Visual Query System over Ontologies for Industry. *Semantic Web*, 1 (1570):1–28, aug 2016.

[165] Ahmet Soylu, Martin Giese, Rudolf Schlatte, Ernesto Jimenez-Ruiz, Evgeny Kharlamov, Özgür Özçep, Christian Neuenstadt, and Sebastian Brandt. Querying industrial stream-temporal data: An ontology-based visual approach. *Journal of Ambient Intelligence and Smart Environments*, 9(1):77–95, 2017.

[166] Michael Sperberg-McQueen, François Yergeau, Jean Paoli, Tim Bray, and Eve Maler. Extensible Markup Language (XML) 1.0 (Fifth Edition). Technical report, W3C, nov 2008.

[167] Srinivas and J A Harding. A data mining integrated architecture for shop floor control. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(5):605–624, may 2008.

[168] M. Tamang, S.K. and Chandrasekaran. Modeling and optimization of parameters for minimizing surface roughness and tool wear in turning Al / SiCp MMC , using conventional and soft computing techniques. *Advances in Production Engineering & Management*, 10(2):2–5, 2015.

[169] Satu Tamminen and Henna Tiensuu. Advances in Data Mining. Applications and Theoretical Aspects. In *Advances in Data Mining. Applications and Theoretical Aspects*, volume 7987, 2013.

[170] Jonas Tappolet and Abraham Bernstein. Applied temporal RDF: Efficient temporal querying of rdf data with SPARQL. In *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*, volume 5554 LNCS, pages 308–322. Springer, Berlin, Heidelberg, 2009.

[171] Walter Terkaj, Giulia Pedrielli, and Marco Sacco. Virtual factory data model. In *Workshop on Ontology and Semantic Web for Manufacturing (OSEMA 2012)*, pages 29–43, 2012.

[172] Moshe Y. Vardi and Moshe Y. A logical revolution (keynote). In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering - ESEC/FSE 2013*, page 1, New York, New York, USA, 2013. ACM Press.

[173] VDI. VDI 5600- fertigungsmanagementsysteme (manufacturing execution systems - mes). Technical report, Verein Deutscher Ingenieure, Düsseldorf, Germany, 2012.

[174] A. Vijayaraghavan and D. Dornfeld. Automated energy monitoring of machine tools. *CIRP Annals - Manufacturing Technology*, 59(1):21–24, jan 2010.

[175] R Volk. *Rauheitsmessung: Theorie und praxis*. PhD thesis, 2018.

[176] Guofeng Wang, Zhiwei Guo, and Yinwei Yang. Force sensor based online tool wear monitoring using distributed Gaussian ARTMAP network. *Sensors and Actuators A*, 192:111–118, apr 2013.

[177] Ke Sheng Wang. Towards zero-defect manufacturing (ZDM)-a data mining approach. *Advances in Manufacturing*, 1(1):62–74, 2013.

[178] Edward O. Wilson. *Consilience: The unity of knowledge*, volume 15. Vintage Books, 1999.

[179] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. Ontology-based data access: A survey. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 5511–5519, 2018.

[180] Hae Sung Yoon, Jang Yeob Lee, Min Soo Kim, and Sung Hoon Ahn. Empirical power-consumption model for material removal in three-axis milling. *Journal of Cleaner Production*, 78:54–62, sep 2014.

[181] Chih Min Yu, Chen Fu Chien, and Chung Jen Kuo. Exploit the Value of Production Data to Discover Opportunities for Saving Power Consumption of Production Tools. *IEEE Transactions on Semiconductor Manufacturing*, 30(4):345–350, 2017.

[182] Jianbo Yu. Machine Tool Condition Monitoring Based on an Adaptive Gaussian Mixture Model. *Journal of Manufacturing Science and Engineering*, 134(3), 2012.

[183] Guojun Zhang, Jian Li, Yuan Chen, Yu Huang, Xinyu Shao, and Mingzhen Li. Prediction of surface roughness in end face milling based on Gaussian process regression and cause analysis considering tool vibration. *International Journal of Advanced Manufacturing Technology*, 75 (9-12):1357–1370, 2014.

[184] Hao Zhang, Hongzhi Wang, Jianzhong Li, and Hong Gao. A generic data analytics system for manufacturing production. *Big Data Mining and Analytics*, 1(2):160–171, 2018.

Appendix A

Digital Appendix

For this thesis, a digital appendix was created. It can be found here: https://github.com/BMoer/Dissertation_Digital_Appendix

Appendix B

Ontop-temporal SQL Unfolding

```
WITH "temporalview_1" AS (
SELECT CAST('ID01' AS VARCHAR) AS "b_id", CAST('TRUE' AS Boolean) AS "
    standby_start_inc",
          "boreholes"."process_stop" AS "standby_start",
          LEAD("boreholes"."process_start",1) OVER(ORDER BY "boreholes"."
    process_start") AS "standby_stop",
          CAST('TRUE' AS Boolean) AS "standby_stop_inc"
          FROM "meta_data"."boreholes"
          WHERE "boreholes"."process_start" IS NOT NULL AND "boreholes"."
    process_stop" IS NOT NULL),
"C_0" AS (
SELECT "b_idf0", "bInc", "b", "e" + INTERVAL '30' SECOND AS "e", "eInc"
FROM (SELECT "b_id" AS "b_idf0", "standby_start_inc" AS "bInc", "
    standby_start" AS "b", "standby_stop" AS "e", "standby_stop_inc" AS "eInc
    "
FROM "temporalview_1") AS "t"
WHERE "e" - INTERVAL '30' SECOND >= "b"),
"temporalview_0" AS (
SELECT CAST('ID01' AS VARCHAR) AS "b_id", CAST('TRUE' AS Boolean) AS "
    process_start_inc", "boreholes"."process_start", "boreholes"."
    process_stop",
          CAST('TRUE' AS Boolean) AS "process_stop_inc"
          FROM "meta_data"."boreholes"
          WHERE "boreholes"."process_start" IS NOT NULL AND "boreholes"."
    process_stop" IS NOT NULL),
"C_1" AS (
SELECT "b_idf0f2", "bIncf3", "bf4","ef5" + INTERVAL '5' SECOND AS "ef5", "
    eIncf6"
FROM (SELECT "b_id" AS "b_idf0f2", "process_start_inc" AS "bIncf3", "
    process_start" AS "bf4", "process_stop" AS "ef5", "process_stop_inc" AS "
    eIncf6"
FROM "temporalview_0") AS "t2"
WHERE "ef5" - INTERVAL '5' SECOND >= "bf4"),
```

```
23 | "temporalview_2" AS (
24 | SELECT CAST('ID01' AS VARCHAR) AS "b_id", "amp", CAST('TRUE' AS Boolean) AS "
       phase_start_inc",
25 |            "phases"."phase_start", "phases"."phase_stop",
26 |            CAST('TRUE' AS Boolean) AS "phase_stop_inc"
27 |            FROM "meta_data"."phases"
28 |            WHERE "phases"."phase_start" IS NOT NULL AND "phases"."phase_stop"
       IS NOT NULL),
29 | "C_2" AS (
30 | SELECT "b_idf0f2f7", "v", "bIncf3f8", "bf4f9", "ef5f10" + INTERVAL '2' SECOND
        AS "ef5f10", "eIncf6f11"
31 | FROM (SELECT *
32 | FROM (SELECT "b_id" AS "b_idf0f2f7", "amp" AS "v", "phase_start_inc" AS "
       bIncf3f8", "phase_start" AS "bf4f9", "phase_stop" AS "ef5f10", "
       phase_stop_inc" AS "eIncf6f11"
33 | FROM "temporalview_2") AS "t5"
34 | WHERE "v" > 0.0) AS "t6"
35 | WHERE "ef5f10" - INTERVAL '2' SECOND >= "bf4f9"),
36 | "C_3" AS (
37 | SELECT "C_1"."b_idf0f2", CASE WHEN "C_1"."bf4" > "C_2"."bf4f9" AND "C_2"."
       ef5f10" > "C_1"."bf4" THEN "C_1"."bIncf3" WHEN "C_2"."bf4f9" > "C_1"."bf4
       " AND "C_1"."ef5" > "C_2"."bf4f9" THEN "C_2"."bIncf3f8" WHEN "C_1"."bf4"
       = "C_2"."bf4f9" THEN "C_1"."bIncf3" AND "C_2"."bIncf3f8" ELSE "C_1"."
       bIncf3" AND "C_2"."bIncf3f8" END AS "bIncf3", CASE WHEN "C_1"."bf4" > "
       C_2"."bf4f9" AND "C_2"."ef5f10" > "C_1"."bf4" THEN "C_1"."bf4" WHEN "C_2
       "."bf4f9" > "C_1"."bf4" AND "C_1"."ef5" > "C_2"."bf4f9" THEN "C_2"."bf4f9
       " WHEN "C_1"."bf4" = "C_2"."bf4f9" THEN "C_1"."bf4" ELSE "C_1"."bf4" END
       AS "bf4", CASE WHEN "C_1"."ef5" < "C_2"."ef5f10" AND "C_1"."ef5" > "C_2
       "."bf4f9" THEN "C_1"."ef5" WHEN "C_2"."ef5f10" < "C_1"."ef5" AND "C_2"."
       ef5f10" > "C_1"."bf4" THEN "C_2"."ef5f10" WHEN "C_1"."ef5" = "C_2"."
       ef5f10" THEN "C_1"."ef5" ELSE "C_1"."ef5" END AS "ef5", CASE WHEN "C_1"."
       ef5" < "C_2"."ef5f10" AND "C_1"."ef5" > "C_2"."bf4f9" THEN "C_1"."eIncf6"
        WHEN "C_2"."ef5f10" < "C_1"."ef5" AND "C_2"."ef5f10" > "C_1"."bf4" THEN
       "C_2"."eIncf6f11" WHEN "C_1"."ef5" = "C_2"."ef5f10" THEN "C_1"."eIncf6"
       AND "C_2"."eIncf6f11" ELSE "C_1"."eIncf6" AND "C_2"."eIncf6f11" END AS "
       eIncf6"
38 | FROM "C_1",
39 | "C_2"
40 | WHERE "C_1"."b_idf0f2" = "C_2"."b_idf0f2f7" AND ("C_1"."bf4" > "C_2"."bf4f9"
       AND "C_2"."ef5f10" > "C_1"."bf4" OR "C_2"."bf4f9" > "C_1"."bf4" AND "C_1
       "."ef5" > "C_2"."bf4f9" OR "C_1"."bf4" = "C_2"."bf4f9") AND ("C_1"."ef5"
       < "C_2"."ef5f10" AND "C_1"."ef5" > "C_2"."bf4f9" OR "C_2"."ef5f10" < "C_1
       "."ef5" AND "C_2"."ef5f10" > "C_1"."bf4" OR "C_1"."ef5" = "C_2"."ef5f10")
       ),
41 | "C_4" AS (
42 | SELECT "b_idf0f2", "bIncf3", "bf4", "ef5", "eIncf6", "bIncf3" AS "bIncf30", "
       bf4" + INTERVAL '5' SECOND AS "bf40", "ef5" AS "ef50", "eIncf6" AS "
       eIncf60"
```

```
43 | FROM "C_3"),
44 | "C_5" AS (
45 | SELECT "C_0"."b_idf0", CASE WHEN "C_0"."b" > "C_4"."bf4" AND "C_4"."ef5" > "
       C_0"."b" THEN "C_0"."bInc" WHEN "C_4"."bf4" > "C_0"."b" AND "C_0"."e" > "
       C_4"."bf4" THEN "C_4"."bIncf3" WHEN "C_0"."b" = "C_4"."bf4" THEN "C_0"."
       bInc" AND "C_4"."bIncf3" ELSE "C_0"."bInc" AND "C_4"."bIncf3" END AS "
       bInc", CASE WHEN "C_0"."b" > "C_4"."bf4" AND "C_4"."ef5" > "C_0"."b" THEN
        "C_0"."b" WHEN "C_4"."bf4" > "C_0"."b" AND "C_0"."e" > "C_4"."bf4" THEN
       "C_4"."bf4" WHEN "C_0"."b" = "C_4"."bf4" THEN "C_0"."b" ELSE "C_0"."b"
       END AS "b", CASE WHEN "C_0"."e" < "C_4"."ef5" AND "C_0"."e" > "C_4"."bf4"
        THEN "C_0"."e" WHEN "C_4"."ef5" < "C_0"."e" AND "C_4"."ef5" > "C_0"."b"
       THEN "C_4"."ef5" WHEN "C_0"."e" = "C_4"."ef5" THEN "C_0"."e" ELSE "C_0"."
       e" END AS "e", CASE WHEN "C_0"."e" < "C_4"."ef5" AND "C_0"."e" > "C_4"."
       bf4" THEN "C_0"."eInc" WHEN "C_4"."ef5" < "C_0"."e" AND "C_4"."ef5" > "
       C_0"."b" THEN "C_4"."eIncf6" WHEN "C_0"."e" = "C_4"."ef5" THEN "C_0"."
       eInc" AND "C_4"."eIncf6" ELSE "C_0"."eInc" AND "C_4"."eIncf6" END AS "
       eInc"
46 | FROM "C_0",
47 | "C_4"
48 | WHERE "C_0"."b_idf0" = "C_4"."b_idf0f2" AND ("C_0"."b" > "C_4"."bf4" AND "C_4
       "."ef5" > "C_0"."b" OR "C_4"."bf4" > "C_0"."b" AND "C_0"."e" > "C_4"."bf4
       " OR "C_0"."b" = "C_4"."bf4") AND ("C_0"."e" < "C_4"."ef5" AND "C_0"."e"
       > "C_4"."bf4" OR "C_4"."ef5" < "C_0"."e" AND "C_4"."ef5" > "C_0"."b" OR "
       C_0"."e" = "C_4"."ef5")),
49 | "view_1" AS (
50 | select CAST('ID01' AS VARCHAR) AS "b_id", "f_all"
51 | from (
52 | select "fitnesses_inner"."borehole", min("f_roundness") as "f_roundness", min
       ("f_roughness") as "f_roughness", min("f_burr_out") as "f_burr_out",
53 | min(("f_roundness"+"f_roughness"+"f_burr_out")/3) as "f_all"
54 | from (
55 | select "qualitycards"."borehole",
56 | case when "roundness_mean" is not null
57 | then (0.06-"roundness_mean")/0.06*100
58 | else null end as "f_roundness",
59 | case when "roughness_ra_mean" is not null
60 | then (1.6-"roughness_ra_mean")/1.6*100
61 | else null end as "f_roughness",
62 | case when "burr_height_in" is not null
63 | then case
64 | when "material"='Aluminum'
65 | then (0.127-"burr_height_in")/0.127*100
66 | when "material"='Titanium'
67 | then (0.2032-"burr_height_in")/0.2032*100
68 | else 0 end
69 | else null end as "f_burr_in",
70 | case when "burr_height_out" is not null
71 | then case
```

```
when "material"='Aluminum'
then (0.127-"burr_height_out")/0.127*100
when "material"='Titanium'
then(0.2032-"burr_height_out")/0.2032*100
else 0 end
else null end as "f_burr_out"
from "meta_data"."qualitycards"
join "meta_data"."layers" on "layers"."layer_id"="qualitycards"."layer") as "fitnesses_inner"
group by "borehole") as "fitnesses_outer"
join "meta_data"."qualitycards" on "fitnesses_outer"."borehole"="qualitycards"."borehole"
join "meta_data"."layers" on "layers"."layer_id"="qualitycards"."layer"
join "meta_data"."boreholes" on "fitnesses_outer"."borehole"="boreholes"."borehole_id"),
"view_0" AS (
select CAST('ID01' AS VARCHAR) AS "b_id" from "meta_data"."boreholes"
join "meta_data"."phases" on "meta_data"."phases"."borehole"="boreholes"."borehole_id")

SELECT DISTINCT 1 AS "bhType", NULL AS "bhLang", 'http://ontologies.ift.at/production_systems.ttl/0.5/Borehole/' || "C_5"."b_idf0" AS "bh", 6 AS "fType", NULL AS "fLang", "t14"."f_allm1" AS "f", 1 AS "bIncType", NULL AS "bIncLang", "C_5"."bInc", 1 AS "bType", NULL AS "bLang", "C_5"."b", 1 AS "eType", NULL AS "eLang", "C_5"."e", 1 AS "eIncType", NULL AS "eIncLang", "C_5"."eInc"
FROM "C_5",
(SELECT "b_id" AS "b_idf0f0", "f_all" AS "f_allm1"
FROM "view_1") AS "t14",
(SELECT "b_id" AS "b_idf0f1"
FROM "view_0") AS "t15"
WHERE "C_5"."b_idf0" = "t14"."b_idf0f0" AND "C_5"."b_idf0" = "t15"."b_idf0f1"
 AND "t14"."f_allm1" IS NOT NULL
```

Listing B.1 SQL Unfolding generated by Ontop-temporal

Appendix C

Curriculum Vitae

Benjamin **Mörzinger**

Alliiertenstraße 9/11, 1020 Vienna, Austria

(+43) 664/89 666 85 | b.moerzinger@gmail.com | bmoerzin

Skills

Programming Python, MATLAB, SQL, OWL, SPARQL, HTML, VBA, LaTeX
Engineering Production Systems, Industrial Communication, Data Analysis, Thermodynamics, Energy Efficiency, Cryptocurrencies
Languages German, English

Experience

Institute for Production Engineering and Laser Technology, TU Wien
Vienna, Austria

Researcher, Project Manager — *Jan. 2017- Jun. 2019*

- Project management FFG flagship project Balanced Manufacturing
- Implementation of a large-scale, open source data monitoring and analysis framework for the manufacturing industry
- Several smaller projects in the field of data analysis and energy efficiency

Institute for Production Engineering and Laser Technology, TU Wien
Vienna, Austria

Researcher — *Apr. 2014- Jan. 2017*

- Modeling, simulation and optimization of production systems
- Development of a large-scale, open source data monitoring and analysis framework for the manufacturing industry
- Lecture: Energy efficiency in production systems

AIT Austrian Institute of Technology GmbH
Vienna, Austria

Diploma Student — *Sep. 2014- Apr. 2015*

- Modeling and simulation of thermodynamic processes relevant for latent heat storage
- Design and assembly of a latent heat energy storage
- Setup of experiments, characterization of the storage and validation of the models

Institute for Engineering Design and Logistics Engineering, TU Wien
Vienna, Austria

Academic Tutor — *Jul. 2013- Apr. 2015*

- Guiding Students through a mechanical design project
- Presentation of solution methods
- Technical evaluation and correction of student projects

OMV AG
Vienna, Austria

Trainee Corporate Procurement — *Aug. 2012 and Aug. 2013- Sep. 2013*

- Development and implementation of a vendor database
- Analysis and technical evaluation of offers
- Documentation of closed tenders

ECOP Technologies GmbH
Vienna, Austria

Engineer — *Jan. 2011- Dec. 2012*

- Design and manufacturing of mechanical prototypes
- Monitoring and design of experimental setups
- Documentation and interpretation of results

Education

Free University of Bolzano
Bolzano, Italy

Visiting Researcher — *Oct. 2018- Mar. 2019*

- Close cooperation with Optique development team
- Research on requirements towards ontology-based data access from the manufacturing domain
- Work on ontology-based access to time-series data

TU Wien
Wien, Austria

PhD. in Mechanical Engineering — *Sep. 2015- Jul. 2019*

- Optimization of production processes
- Supervision of several bachelor and master thesis
- PhD Thesis: Accessing Manufacturing Data through Virtual Knowledge Graphs

TU Wien

Wien, Austria

MSc. in Mechanical Engineering

Sep. 2013- Sep. 2015

- Energy management
- InnoLab- Development of novel safety measures for motorcyclists (in cooperation with KTM)
- Diploma thesis: Design, erection and measurements on a latent heat energy storage unit on the basis of macro encapsulated polymer

University of Strathclyde

Glasgow, United Kingdom

Erasmus

Jan. 2013 - Jun. 2013

- Project: Participation in a ESA- tender
- Advanced methods of fluid dynamics
- Societies: boxing and hiking

TU Wien

Wien, Austria

BSc. in Mechanical Engineering

Okt. 2009- Sep. 2013

- Lightweight structures
- Integrative product creation
- Bachelor thesis: Project management methods for small scale projects

Extracurricular Activity

Pfadfinder und Pfadfinderinnen Östereichs

Austria

Strategic Coordinator

Apr. 2018 - PRESENT

- Team lead for strategic management of change process on national level
- Emphasis on new methods to incorporate voluntary work in organisations and better acceptance in industry
- Development of methods and tools for coordinating work of distributed teams

Pfadfinder und Pfadfinderinnen Östereichs

Austria

Team Leader

Nov. 2014 - PRESENT

- Responsible for a team of six voluntary members
- Coaching and training of 13-16 year old teenagers
- Organisation of weekly meetings and camps

Johanniter

Austria

Paramedic

Jan. 2011 - Dez. 2011

- First aid assignments
- Communication and coordination with hospital personnel
- Transport of elderly people

Presentations

IOT Forum

Vienna, Austria

Presenter for <IOTA- Theory and Application>

May 2018

- Application of IOTA cryptocurrency in the field of production engineering

Innovationsforum Energieeffizienz in der Produktion

Linz, Austria

Presenter for <Balanced Manufacturing>

Nov. 2017

- Presentation of research project BaMa- Balanced Manufacturing

Science Slam Vienna

Vienna, Austria

Presenter for <Maschinen verstehen - Wie aus Funken Wissen wird>

Nov. 2016

- Scientific stage performance