MASTERARBEIT / MASTER’S THESIS

Titel der Masterarbeit / Title of the Master’s Thesis

Automatic Extraction of Dimensioning Requirements from Engineering Drawings

verfasst von / submitted by

Beate Scheibel, BSc (WU) BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2020 / Vienna 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 926
Studienrichtung lt. Studienblatt / field of study as it appears on the student record sheet:	Masterstudium Wirtschaftsinformatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Math. oec.Dr. Stefanie Rinderle-Ma

Declaration of Authorship

I, Beate Scheibel, BSc (WU) BSc, declare that this thesis titled, "Automatic Extraction of Dimensioning Requirements from Engineering Drawings" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I tried to be as clear as possible to describe what was done by others and what I have contributed myself.
- Where the thesis is based on work done by myself jointly with others, the concepts and ideas always are my contribution. Implementation details are sometimes contributed by others, if so, this fact is clearly stated and their contribution is acknowledged.

Signed: ____

Date: ____

UNIVERSITY OF VIENNA

Abstract

Faculty of Computer Science
Workflow Systems and Technology Group

Automatic Extraction of Dimensioning Requirements from Engineering Drawings

by Beate Scheibel, BSc (WU) BSc

Even though the process of constructing and manufacturing a workpiece gets more and more automated, the design and use of technical drawings is still not fully integrated in the automated production process. Technical drawings accompany the workpiece throughout its life cycle and are important for additional information which is typically not part of the CAD-model, for example tolerances and regulatory standards. Goal of this master thesis is to provide a system which is able to extract textual information from technical drawing in order to process this data and automatically generate a user interface which can be used as support for measurement and quality control of workpieces. This is done by extracting textual data from the drawings, clustering these using DBSCAN and post-processing these elements to get essential dimensioning requirements. Additionally, relevant regulatory standards are extracted and added to the user interface to provide employees with all necessary information to measure and assess the quality of a given workpiece. An evaluation of the prototype shows that overall more than 70% of relevant dimensions can be extracted, whereby this value is dependent on the complexity and layout of the drawings. In addition, design guidelines are developed to optimize the readability of drawings for machines as well as for humans and additional application scenarios are explored.

UNIVERSITÄT WIEN

Kurzbeschreibung

Faculty of Computer Science
Workflow Systems and Technology Group

Automatic Extraction of Dimensioning Requirements from Engineering Drawings

von Beate Scheibel, BSc (WU) BSc

Industrie 4.0, Automatisierung und Vernetzung von Maschinen und Produktionsanlagen ist derzeit ein viel erforschtes Gebiet. Die Produktionskette soll möglichst durchgängig automatisiert werden um Ressourcen einzusparen und die Qualität zu erhöhen. Ein Bestandteil dieser Produktionskette, welchem jedoch weniger Beachtung geschenkt wurde, ist die Integration von technischen Zeichnungen in eine automatisierte Prozesskette. Technische Zeichnungen beinhalten meist Informationen die nicht in im CAD-Modell zu finden sind und begleiten das Werkstück durch den gesamten Produktionsprozess. Ziel dieser Masterarbeit ist, ein System zu entwickeln, welches automatisch Informationen aus technischen Zeichnungen ausliest, diese dann verarbeitet und daraus eine Benutzeroberfläche generieren kann. Diese soll es Nutzern erleichern ein Werkstück auszumessen und dessen Qualität zu kontrollieren. Im Zuge dieser Arbeit werden alle textuellen Bestandteil aus einer Zeichnung extrahiert, welche dann mithilfe von DBSCAN gruppiert und anschließend nachbearbeitet werden, um nur relevante Bemaßungen und Toleranzen anzuzeigen. Zusätzlich werden Daten ausgelesen, die Hinweise auf die anwendbaren Normen geben. Diese Standards werden ebenfalls auf der Benutzeroberfläche angezeigt um den Mitarbeitern alle Informationen zur Verfügung zu stellen, die benötigt werden um das Werkstück zu kontrollieren. Eine Evaluation dieses Prototypen zeigt, dass durchschnittlich über 70% aller relevanten Bemaßungen erkannt werden. Dieser Prozentsatz hängt jedoch stark von der Komplexität sowie dem Aufbau der Zeichnung ab. Zusätzlich dazu werden Richtlinien beschrieben, welche die manuelle als auch automatisierte Lesbarkeit von technischen Zeichnungen erleichtern sollen, während des Weiteren alternative Anwendungsfälle beschrieben werden.

Acknowledgements

I want to thank Prof. Dr. Stefanie Rinderle-Ma and Jürgen for providing me with continuous feedback and pushing me to finish this thesis. Additionally, I want to thank Dr. Christoph Pollak for providing me with the opportunity to write this thesis as part of my employment at the CDP. Finally, I am extremely grateful to my family for always supporting me.

And of course, I want to thank all doggos for being fluffy and cute and providing me with online content to help me procrastinate.

Contents

List of Figures

List of Tables

Chapter 1

Introduction & Motivation

The continuous progress in technology saw a rise in automation in all areas of life, including the production and manufacturing domain. "Smart production", "smart factory" and "industry 4.0" are commonly used buzzwords, not only in the scientific community but in production and industry as well. Machines tend to get smarter and therefore more efficiently integrated in virtual networks, while process optimization and automation is done at each level of the production process, cyber-physical systems can increasingly collaborate and cloud computing offers new possibilities (for example [1] or [2]). However, there are still some basic issues on the shop floor that need to be tackled before a higher level of automation and integration can be attempted.

A simplified exemplary production process chain includes the drawing of a model by the means of a CAD (computer-aided design) program, well-known representatives of those programs being AutoCAD or Solidworks. This model is then transformed via CAM (computer aided manufacturing) - tools for example Catio or Esprit into an NC (numerical control) program, which is in turn used by a tooling machine to manufacture the desired workpiece. Designing and production processes can either be seen as two separate steps or can be developed together using an integrated CAD-CAM system. However, this process is not "smart" yet and still involves a lot of human labor and input. There are multiple approaches for automating the production chain on various levels including automatic generation of CAD-models or better integration of CAD-CAM systems. Having said that, one essential step that was omitted in the above description of the production process, and still needs optimization regarding integration in the production chain, is the creation of an engineering drawing. Engineering drawings, also called technical drawings, are 2D depictions of the requested workpiece that include additional textual information such as the exact measurements, tolerances, applicable

ISO (International Standardization Organization) norms and more, which are essential for quality control of the finished workpiece.

Engineering drawings are an essential part of the production and development of different products in a multitude of industries. Nowadays 3D CAD-models are typically used for the actual production process. Nevertheless, engineering drawings are still widely used [3] for other purposes. According to [4] 250 million new drawings are generated each year and the U.S. Army alone has tens of thousands of legacy engineering drawings stored. These are still mostly applied for the contractual basis and accompany the workpiece throughout the product life cycle [5]. Additionally, the textual information is mainly noted on the drawing as the CAD-model solely includes the nominal values. Having said that, there is still no solution available that is able to automatically read information out of engineering drawings. In fact, the transformation from drawings to CAD-models and the extraction of information has been a well researched topic for the last decades. However, there are still no satisfying results (see for example: [6], [3], [7] and [8]). [6] gives a review over the current literature regarding the digitalization of engineering drawings, reaching the conclusion that still more work needs to be done in this field of research. There are approaches to include all of the additional information regarding dimensioning and tolerances in the CAD-model (for example PMIs, product and manufacturing information, see for example [9]), which are designed to solve this gap in the process. However, it is still common practice to only include this kind of information in technical drawings.

Figure 1.1 depicts the aforementioned issues. Even though, the transformation from CAD-model to NC program can be done (almost) seamlessly, the technical drawing is not integrated efficiently.

A solution that would allow the digitalization of engineering drawings could be used for making use of older drawings, where no respective CAD-model exists, as well as extracting information that is not included in the CAD-model, such as tolerances. This additional information could then be used to automate the entire production process including measuring and quality control. An optimal solution should be able to extract all available information including graphical elements as well as meta-data like the dimensioning requirements [10]. However, it is not always necessary to extract geometric and graphical elements, as an additional CAD-model exists in a lot of cases. The problem is to include additional textual information in the process to create a seamless production chain. This additional information refers not only to the dimensions and

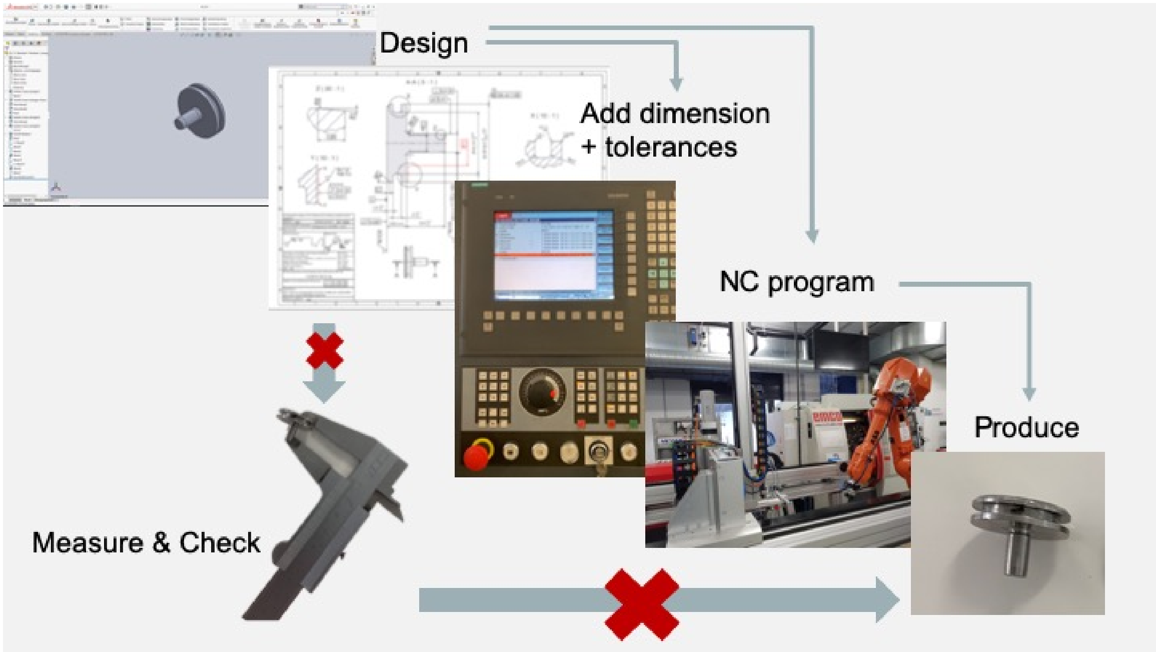


Figure 1.1: Process of Designing and Manufacturing a Workpiece.

tolerances written on the drawing itself but also the information that is part of the regulatory framework e.g. ISO or DIN (Deutsches Institut für Normung) standards. These regulatory documents usually refer to minimum standards that should be satisfied or specify the default dimensioning requirements, if these are not stated explicitly in the drawing.

To sum up, there is still no solution of integrating the textual information that is found in technical drawings into a continuous (semi-)automated production process. This issue is relevant as a solution would facilitate automating different additional processes for example quality control by the means of automated measurement. Goal of this master thesis is therefore to create a solution for a sub-part of the textual extraction, in particular the extraction of dimensions and the related tolerances and prepare these parameters in a human and machine-readable form that can be readily used in a production environment.

This issue was discovered as part of the authors work at the ACDP (Austrian Center for Digital Production) [1], which does research in the areas of digitalization and automatization and works closely with the "Pilotfabrik" [2] by the technical university of Vienna, where theoretical research can be applied in real life production contexts.

[1] ACDP `https://www.acdp.at/`

[2] Pilotfabrik `http://pilotfabrik.tuwien.ac.at/`

1.1 Research Questions

The preceding problem description leads to a more precise definition of research questions that are the focus of this master thesis. In total two main research questions are addressed in this work, each one having two sub-questions.

First Research Question
How can textual information, specifically dimensioning requirements, be extracted out of an engineering drawing?
Sub-question 1a
How does the extracted information relate to regulatory documents?
Sub-question 1b
How can relevant information from regulatory documents be identified and extracted?

The first research question addresses specifically the issues discussed in the introduction. To answer these questions a literature search will be conducted and a prototype will be developed. The second research question then focuses on analyzing the prototype, especially in the context of production.

Second Research Question
In which ways is this prototype relevant to the production domain?
Sub-question 2a
How well can the extraction be achieved in terms of accuracy?
Sub-question 2b
Which alternative application scenarios are possible?

These questions focus on assessing the usability of the prototype in the industry as well as establishing some key figures to analyze the accuracy of the extraction process and establish further use cases. These performance indicators can the be used to further optimize the prototype in future work.

1.2 Research Methodology

The used methodology for this thesis is ”design science”. Design science is defined by creating new knowledge through the systematic building of artifacts and is frequently used in information science [11]. This methodology wants to unite a purely theoretical approach with the practical solution of a real problem. [12] describe it as learning new knowledge through the act of building an artifact. The basic approach of design science is to implement and subsequently test and evaluate a developed prototype. The results of the evaluation are used to adapt and optimize the implementation. This leads to multiple iterations of developing and testing a solution. This process of development and testing should in turn also lead to the creation of new theoretical knowledge and the refinement of existing theory. [12]

Figure 1.2[12] depicts the typical process for a design science project. The iterations are clearly displayed by the arrows on the left hand side.

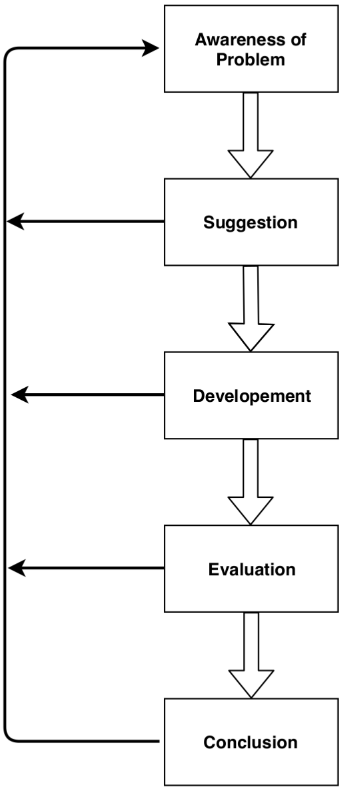


Figure 1.2: Suggested Workflow for a Design Science Project.

In this case, knowledge about extraction of information from engineering drawings is created by solving the actual problem through the implementation of a solution. Relating the design science steps to this thesis, the first phase was already covered in the problem discussion in the previous subsection, additionally the issue will be further discussed in section 2, so the "Awareness of Problem" phase is completed. The "Suggestion" phase consists of hypothesizing about a solution. In this thesis it was already briefly discussed as well beforehand. However, there will be a more in-depth-discussion of possible problem solutions in chapter3. The chapters 4 and 5 correspond to the respective phases of implementation and evaluation in the design science methodology. The last section (section 6) corresponds to the conclusion phase.

1.3 Contributions

The contributions of this thesis can be divided into contributions to the scientific community and to the domain.

Contributions to the scientific community
This thesis includes an approach for extracting dimensioning requirements from engineering drawings as well as an evaluation of its practicability. Additionally, different approaches to achieve the extraction are tested and analyzed. A further contribution is the detailed assessment of how regulatory documents are connected to engineering drawings and how information contained in these documents can help to improve the readability for humans.

Contributions to the domain
As described in the expert interview (see section 5), the presented approach can be used for simplifying quality control. Additionally, the developed prototype can be used to determine the quality of engineering drawings in regard to the readability of the drawing for humans as well as for machines. This can be seen as feedback for design engineers. Furthermore, in section 6 some guidelines are establish that can help further improve the readability of engineering drawings.

1.4 Structure of the thesis

The subsequent chapter 2 consists of a description of the essential concepts as well as a literature review regarding existing approaches for the aforementioned research questions. Chapter 3 describes possible solution designs including a brief requirement analysis. Chapter 4 includes a detailed description of the prototype and corresponding design decisions. A quantitative and qualitative evaluation of the developed prototype is covered in section 5. The remaining section covers the conclusion (section 6) including limitations and ideas for further research on this topic. In the appendix a detailed description on how to start and run the prototype can be found.

Chapter 2

Concepts & Related Work

Before an analysis of related literature specific to the aforementioned issues, some necessary concepts are introduced and briefly discussed. To begin with, some fundamentals of technical drawings, the basis for this thesis are described, including an introduction to dimensioning and tolerances used in technical drawings. In addition, regulatory documents, relevant for these drawings, are outlined briefly. After these essential concepts are covered, a concise summary of the available literature regarding information extraction from technical drawings is given.

2.1 Technical Drawings

Technical drawings are a means to specify the requirements for a certain workpiece and are used as a universal language between engineers, machine operators and generally a wide range of employees that are involved in design and production processes. Technical drawings are used in a multitude of domains reaching from architecture to electrical engineering, with varying symbols and conventions. The information contained in these drawings includes that about geometry, dimensions, tolerances, material, finishes (e.g. surface quality), organizational information and more [13]. Even though standardized guidelines (see subsection 2.2) for constructing such drawings exist, the drawings themselves can vary considerably from one another. Figure 2.1 shows an exemplar drawing displaying the different parts it consists of, which will be discussed in more detail.

Technical drawings can be created by hand or using a CAD-program. Standardized formats for exchanging digital drawings are for example DXF, DWG, IGES or STEP [5]. However, it is common to use an image format like TIFF or PDF for storage and inter-company exchange. This subsection covers a brief summary of the basics of

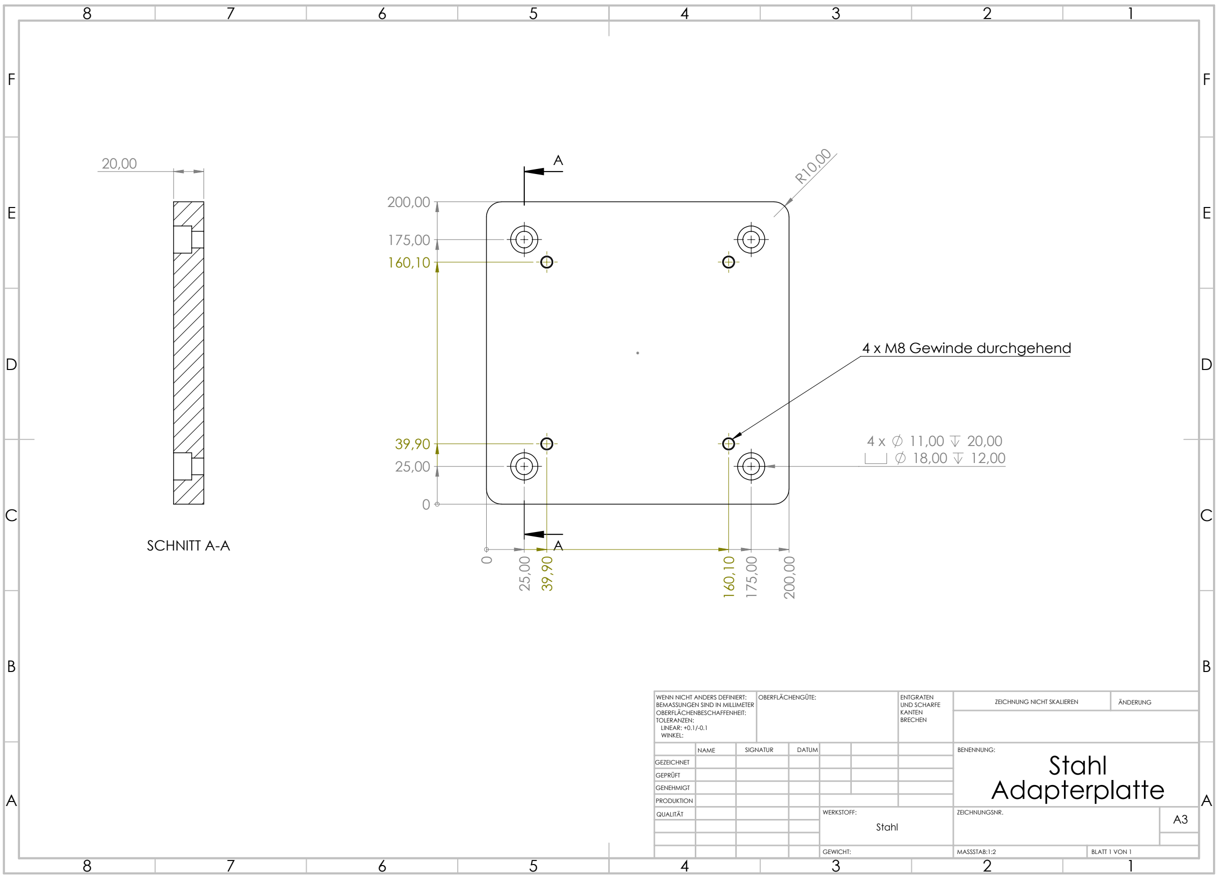
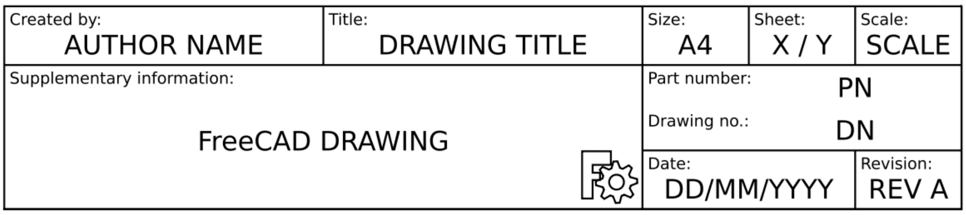


FIGURE 2.1: Example of a Technical Drawing.

technical drawings focusing on the relevant aspects for this thesis. The focus lies on the production domain and in particular on component or part drawings. For a more in-depth introduction see for example [14], [13], [15], [16] and [5].

2.1.1 Structure of engineering drawings

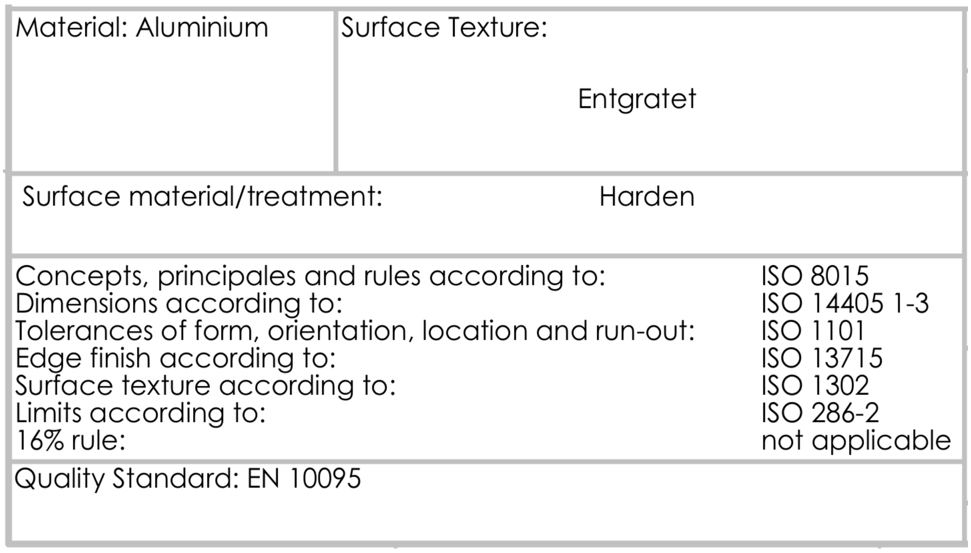
An engineering drawing consists of multiple standardized sections called blocks. Usually the main part consists of the depiction of the workpiece from the main view. Additionally, detail representations of specific sub parts, multiple views of the workpiece or a 3D-representation can be displayed as well. In the right lower corner there should be, according to DIN EN ISO 7200, the title block including textual information like the owner, title, a number for precise identification, and more organizational details. There can be additional tables including a bill of material, a list of all revisions and other additional data. [13] In this thesis only the title block and tables including information about regulatory standards are relevant.



Created by: AUTHOR NAME	Title: DRAWING TITLE	Size: A4	Sheet: X / Y	Scale: SCALE
Supplementary information: FreeCAD DRAWING		Part number: PN Drawing no.: DN		
		Date: DD/MM/YYYY		Revision: REV A

Figure 2.2: Example of a Title Block.

Figure 2.2 shows the title block, whereas figure 2.3 shows an additional table in which all relevant regulatory standards, as well as specific tolerances regarding the surface and the edges are noted. Figure 2.2 is a template title block from ”FreeCAD”, an open-source CAD-tool [17].



Material: Aluminium	Surface Texture: Entgratet
Surface material/treatment:	Harden
Concepts, principales and rules according to:	ISO 8015
Dimensions according to:	ISO 14405 1-3
Tolerances of form, orientation, location and run-out:	ISO 1101
Edge finish according to:	ISO 13715
Surface texture according to:	ISO 1302
Limits according to:	ISO 286-2
16% rule:	not applicable
Quality Standard: EN 10095	

Figure 2.3: Example of Additional Information in a Drawing.

2.1.2 Dimensioning and Tolerances

The dimensions given in these drawings are used as instructions for the machine operator as well as minimal requirements that have to be fulfilled during quality control. Accordingly there are different kinds of dimensioning - function-related, construction-related and inspection-related. However, this just refers to the entry of dimensions in the drawing, not the dimension itself. [13] For this reason, there will be no distinction between these different kinds throughout this thesis. It is not always possible to manufacture the workpiece accurately down to fractures of a millimeter, due to machine or material features, and it is not necessary for each workpiece. The price of manufacturing

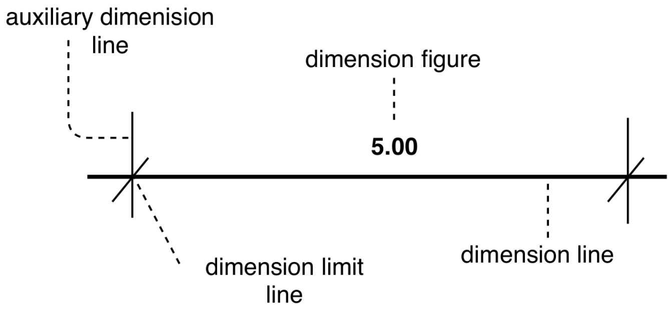


Figure 2.4: Dimensioning and Auxiliary Lines.

is higher the more precise a workpiece has to be, therefore it is desired to construct the workpiece in a way that the measurements are as close as possible to the requirements as needed to ensure reliable function and to be able to function well in conjunction with other parts, but the tolerances are as broad as possible to reduce unnecessary scrap and therefore be as efficient as possible [13].

Dimensioning requirements can cover, among other things, length, height, angle or curve of a geometric object. The dimensions are usually noted directly by the respective graphical element. Additionally, auxiliary lines can be used to specify exactly which structural element the specification refers to.[14] This can be seen in figure 2.4 [14]. The value denoted next to the graphical element is called the nominal dimension, the actual value should lie between the minimum and the maximum tolerance, this area is also called the tolerance zone[5]. Regarding figure **??**, "7,3" is the nominal value, "+0,1" the maximum tolerance which leads to an upper deviation of "7,4" and "-0 "is the minimal tolerance which means that the lower deviation is at "7,3" and the tolerance zone lies between "7,3" and "7,4".

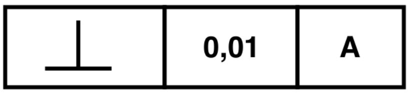
Regarding the tolerances there are a different kinds [5]:

- Dimensioning and size tolerances: These include the evident parameters like the length of a line. Figure 2.5 shows an example.

$$7{,}3\ ^{+0{,}1}_{-0{,}1}$$

Figure 2.5: Example of a Size Tolerance.

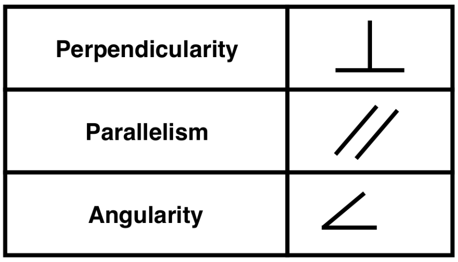
- Geometrical tolerances These tolerances determine how much the form or position of an element can vary, as well as other geometric measurements like orientation and run out. Figure 2.6 is an example for a geometrical tolerance. The symbol means perpendicularity and the dimension would be interpreted as the perpendicularity of the specific element compared to part "A" cannot differ more than "0,01".



⊥	0,01	A

Figure 2.6: Example of a Geometrical Tolerance in a Technical Drawing.

Figure 2.7 gives some examples of symbols used for geometrical tolerances. Additionally, "ø" is used to symbolize a diameter whereas "R" is used to mark a radius, "S" stands for sphere and "M" is used when the element is a thread, that adheres to ISO standards [5].This is not exhaustive, but only includes the symbols that were used in the design and testing process of the prototype, as there is a considerable number of symbols used today. For interpretation and measurement purposes one has to combine the given dimensions and tolerances with the graphical elements and potentially regulatory standards. Therefore it can be a complex task to interpret and apply these drawing correctly.



Perpendicularity	⊥
Parallelism	//
Angularity	∠

Figure 2.7: Some Geometrical Symbols and Their Label.

2.1.3 Regulatory and standardisation documents

As technical drawings serve as a communication medium that is to be understood by everyone working in this sector, it is essential that certain standards are satisfied. There are multiple regulatory frameworks that cover technical drawings. Their goal is not only to unify the language but also to ensure equal quality and safety, therefore facilitate international cooperation and trading. [15] These frameworks standardize basically all features involved in engineering drawings, from symbols to paper sizes to line width and font. The following list contains well known standardization organizations. Each one of them works, among other things, on streamlining production processes, including the design of engineering drawings by developing guidelines and conventions. Most of the conventions overlap, especially since the ISO actively works on unifying the regulations. However there are country or region specific regulations as well. [13]

- ISO: "International Organization for Standardization", there are more than 150 member countries, which work on establishing regulations and norms that apply to all countries and should facilitate international trade. Most regional regulations implement ISO regulations in national law. [15]
- DIN: DIN standards are published by the "Deutsches Institut fuer Normung" and are therefore german standards, often ISO norms are adopted without change, this is shown in the nomenclature as such a standard is titled "DIN ISO". [13]
- EN: "Europaeische Normen", European norms, which have to be translated to national laws first and are then termed DIN EN, or if their are ISO standards as well they could also be named "DIN EN ISO" which means that an ISO norm was implemented in European law and even further implemented in German regulatory works. [13]
- ANSI/ASME: American National Standards Institute, is mainly relevant in the US. However, this institute is also a member of the ISO and therefore adheres to ISO norms and influences new ISO regulations. [15]

The standards used in the drawings as part of this thesis are mainly "ISO", "DIN" and "EN", as "ANSI" typically does not apply to European products.

After this brief overview of technical drawings, the next section covers related literature regarding information extraction out of these documents.

2.2 Information Extraction from Engineering Drawings

This chapter covers related work regarding extraction of information from engineering drawings. Firstly a short introduction to information extraction in general is given, then more specific literature is analyzed.

2.2.1 Information Extraction

The scientific community that deals with information extraction tries to automatically analyze large batches of unstructured text and identify entities, relationships, and in general, extract some structured meaning, out of it. For this purpose often NLP (natural language processing) related techniques (e.g. named entities) as well as machine learning methods are used. [18] Additionally a system to identify entities and their relations has to be constructed, this can be done by manually constructing patterns as well as using machine leaning techniques. [19] The field of information extraction overlaps with several other research fields one of these being text mining. Text mining focuses also on extracting previously unknown information out of textual sources, however the main difference lies in that text mining is more exploratory, whereas information extraction focuses on predefined, specific information. Therefore information extraction can be used as part of text mining and vice versa. Data mining also overlaps but specifies more on extracting information out of databases, XML-files or other structured sources.[19] Knowledge discovery is the super ordinate concept to data mining describing methods for automatic, exploratory analysis and modeling of data. All of these research fields use algorithms and concepts from machine learning, statistical methods and NLP techniques. [20] Figure 2.8[21] shows different related fields and how they interact with each other.

For this thesis, information extraction seems like a suitable approach as it is already known which entities are needed (the dimensions and tolerances), therefore only the actual values have to be extracted, no exploratory search for information is needed and the drawing itself is mainly unstructured, but still adheres to a some basic regulations on the structure.

The literature regarding information extraction in the context of engineering drawings can be divided into two main categories: extracting all information, mainly focusing on the graphical elements, or specifically extracting only (specific) textual information. The next section covers both approaches, however this is only to provide a comprehensive overview and to illustrate the difficulties that are still associated with extracting graphical information out of engineering drawings and therefore solidifies the decision to limit the research questions of this thesis on textual elements only.

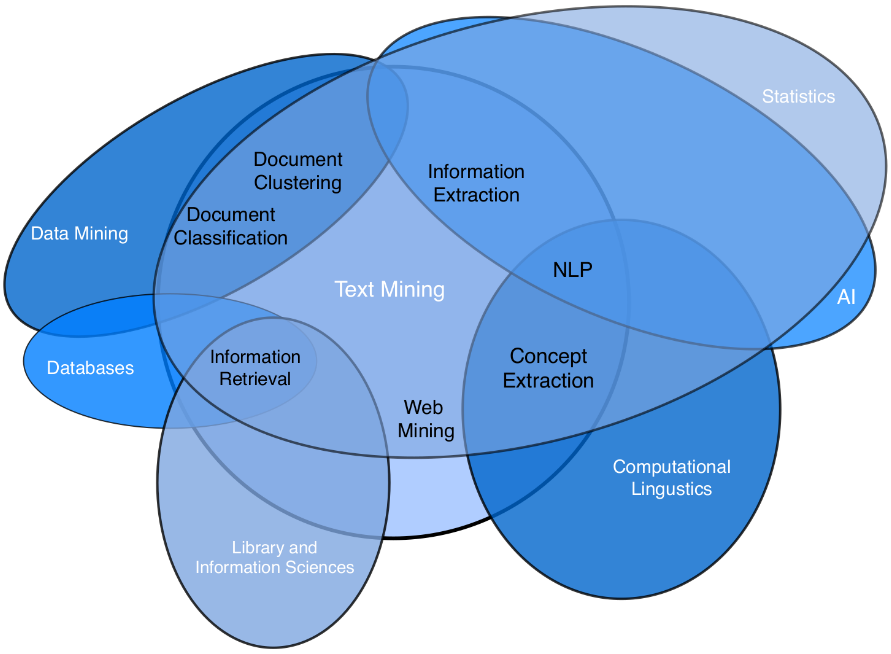


Figure 2.8: Overview of Research Fields Related to Information Extraction.

2.2.2 Extraction of Graphical Elements

This subsection covers work focusing on extracting graphical elements out of the drawing e.g. for creating a 3D-model. For example [22] covers the conversion from a drawing in image format to a CAD model, focusing on distinguishing and extracting symbols, graphical elements and text by using morphology to detect connected components and geometry features like convexity. That this issue has been a focus in the research community for some time is shown by [23], which was written in 1999 and already mentions the problem of converting 2D paper-based drawings into CAD models. The author proposes a system which extracts object lines and dimension sets and assembles 3D objects from vectorised object lines. Likewise, [24], [25], was published in the early to mid 1990s. The authors describe a prototype called Celesstin which should enable the automatic conversion of a drawing into a CAD model by vectorising the drawing and combining it with domain knowledge about the structure, syntax and semantic of the drawing. It is a blackboard-based system that combines multiple approaches for drawing interpretation. They also provide a user interface in which the user can change misinterpreted features. However, the authors come to the conclusion that the knowledge needed for interpretation quickly becomes unmanageable and in addition they do not mention the extraction of dimensions. [26] also uses vectorisation. A system called VEDI is proposed that should be able to automatically recognize engineering drawing entities like lines, arcs, blocks and also dimensions. A binary image is taken as input and transformed to an intermediate vector representation. The object recognition is done by firstly recognising the contours, extracting simple graphic elements, connecting straight lines and assembling these lines. Each already recognized element is then used to detect more elements. The dimensions are recognized by using a knowledge base which is used to search for an initial dimension component, and the parsing the rest in accordance to the already recognized elements. [27] outlines an approach to facilitate retrieving of drawings as well. Their approach consists of a visual classification scheme based on shape geometry and spatial relationships combined with a multidimensional indexing approach. The used features consists of topological relationships. Most other features are removed to reduce the number of elements. Then the remaining elements are divided into a hierarchy of blocks and later transformed into a topology graph while graph descriptors are computed which are then used for a faster way of matching drawings. [28] similarly describes a system for content-based retrieval of vector drawings, that should facilitate the reuse of drawings by providing a drawing sample for the search. This is also done by representing the drawing as a graph of objects and components and reformulating the problem as a graph matching problem. [29] deals with extracting of hand-drawn graphical symbols out of drawings and learning new symbols in the process, also using a graph-matching algorithm. [30] introduces an approach to automatically extract structural information as well as relationships from vector-based drawings. The drawing should firstly be reduced to only the essential information. Subsequently the remaining elements are analyzed in regard to structural information as well as relationships between elements using the concepts of graphical and text primitives in combination with a knowledge base to recognize important symbols. [31] are working on automatic extraction of information of piping and instrumentation diagrams, which are different from the technical drawings discussed in this thesis, however their approach is very interesting as they use deep learning to identify and match the graphical and textual elements.

The following papers all deal with drawings in DXF format, which is commonly used as an exchange format for engineering drawings, to generate a user interface.. A DXF file is written in ASCII and organised by group codes and values. Therefore, extracting information is straightforward, as everything is already organised and machine readable. Accordingly the focus in these papers does not lie on extracting the information itself, but rather organising it or creating a 3D model out of a 2D drawing. [32] extracts information about lines as well as information about the graphical elements itself from a DXF file. The information extracted is subsequently connected to information about these drawings already stored in a database. All this data is then combined to achieve a visual display including additional information. [33] similarly deals with the extraction of geometric elements out of a DXF file by organising the data by group codes. Different features are extracted and labeled. Extracted dimensions are used to calculate the volume and shape of a 3D object and then a 3D model is created. [34] describe their system AUTOFEAT which, likewise, should be able to extract graphical features, manufacturing information as well as information about dimensions and tolerances out of DXF files by using string based pattern recognition and NLP techniques. The DXF file is read and line and arc data is transformed into a directed graph and subsequently organised into different views by differentiating between open and closed loops in the graph to differentiate between the different details. All additional information is assigned to these views by matching the coordinates. Afterwards all loops are converted into string patterns by describing the structural relationships. To extract geometrical and topological features a pattern recognition technique in conjunction with the aforementioned string pattern is used. Likewise [35] describes a method to construct 3D models out of DXF drawings by simulating the human process of of understanding drawings by looking at the whole as well as local details and analysing the relationship between these. The different groups and respective values and features are extracted and matched (e.g. a specific graphical element is matched with its coordinates and all information regarding this element). [36] similarly deals with extracting graphical elements out of drawing in DXF format. [37] also describes the extraction of features out of DXF files and present them in a tabular as well as in graphical form.

Before moving on to the next chapter, one paper needs to be mentioned that can not be clearly assigned to this chapter or the next. [38] covers the thematic of detecting text regions in scanned drawings and separating these from the graphical elements which should lead to an improved extraction of textual information. This is done using OCR on the drawings and then erasing linear components and recognizing other graphical elements by using, among others, stroke density information.

2.2.3 Extraction of Textual Information

Multiple papers took a similar approach to this thesis, extracting specifically textual information. Multiple papers focus on extracting the information out of the title block or additional tables. For example[3] describe their approach to extract knowledge out of tables. Tables are seen as a structured document. First the table section is extracted by dissecting the structure of the document and finding the table layout, then graphical elements and text is recognized and analyzed by using domain knowledge. By comparing physical and logical structure, knowledge should be extracted and then combined into a new standardised table. [39] also extracts information out of tables, focusing on the versioning information to facilitate version management, by using a knowledge-based extraction method by the means of predefined keywords. The drawing is searched for all rectangular regions that contain a group of other rectangulars and specific keywords. All strings comprised therein are then extracted and analyzed if they comply to the naming standards for versioning. If a matching string is found, this string is stored as the versioning information. Otherwise the search is continued. [40] describes an approach facilitating the search for a specific drawing by extracting information out of the title block. This should lead to a more simplified process of reusing drawings to save resources. The automatic search could be achieved by looking for a rectangular or for straight lines constituting the table. Then only this area is further processed. The paper comes to the conclusion that this task is a complex problem due to the variety in drawings, as there can be more than one table, the lines in the tables are not always solid lines and the structure inside the table can vary as well. [41] also tries to extract information out of the title block as well as out of the bill of material by using DXF format. According to the authors this should be achieved by position recognition i.e. obtaining the position of each cell by analyzing the coordinates and only then extracting information. Extracting data from the information tables is also the focus in [42], here in the light of product data management and establishing a connection between the tables as well as the model and the table. This is done by a depth-first search of the tables and the references directly in the DWG file on the AutoCAD platform. [43] focuses on the bill of material and specifically on creating an adequate measure of distance to perform clustering on the BOM (bill of materials) to get families of products. [44] wants to facilitate managing a large amount of connected drawings, by trying to create hyperlinks in the drawings. This should, according to the authors, be accomplished by gathering information about the drawings by locating mentions of other drawings (in this case called anchor shapes which refer to other documents). This is done by recognising the special shape of these anchor regions. After that, OCR is performed to extract the text inside these regions, where the "destination" drawing is mentioned. With that information a connection graph is created where links between all documents of one project are displayed. In total they were able to obtain an accuracy of almost 95% of detection and hyperlinking, although they mention the difficulty of high noise levels in the drawings.

[45] creates connections between CAD models and paper-based drawings by using OCR to extract information about the drawing from the textual information in the tables and title block, storing this information in an ontology, searching the 3D models by string matching and thereby establishing relationships among these. To gather the textual information it is necessary to recognize the tables correctly and detect the subfield, this is solved by creating a "master" file with relative coordinates, to extract the different kinds of information blocks. The authors also come to the conclusion that this is a very complex issue as drawings, even tough standards exist, vary greatly in the use

and position of information blocks. [46] focuses also on OCR of technical drawings, in particular on a classification of machine-written and printed text by using an SVM (support vector machine) in order to simplify the extraction of the respective element.

The following papers deal specifically with extracting dimensions and therefore are the closest to this thesis. [7] describes an approach to extract dimensioning information out of drawings in image format. The approach consists of text segmentation by extracting textboxes out of the drawings. The textboxes can either be a "basic" textbox (containing one continuous string) or a "logical" textbox containing all basic textboxes that belongs together (dimension and tolerances). Everything is reduced to basic textboxes, then OCR is used to obtain the values. Afterwards the values are recomposed to logical boxes including dimensions and the respective tolerances by using heuristics (for example tolerances have to be significantly smaller than the dimension itself). [10] also covers the extraction of dimensions, specifically dimensions adhering to ISO standards. First, text detection is done by separating text and graphic, subsequently the image is vectorised and indicating primitives, essentially candidates for being dimension sets, are extracted by analysing lines, relative position and distance to other elements. These candidates can be divided into atomic primitives and complete dimensions (consisting of atomic primitives), similar to the basic and logical textboxes mentioned above. The combination of atomic primitives is done by proximity and same thickness of the elements. [47] describes an approach to recognize dimensions in vectorized drawings by detecting lines and arrowheads as it is assumed that dimensions are also near a line with one or two arrowheads. So for each of these spotted lines with an arrowhead a search for a rectangular containing text is conducted. And the nearest found text is associated with the lines and corresponding graphical elements. [48] uses a similar approach for detecting dimension sets, in addition to the arrowhead technique, a special rule-based text/graphic separation algorithm was used in which even complex dimension sets should be detected. They also use primitives and matching of these with associated graphical elements. [49] also extract textual information including dimensions by using heuristic procedures including arrowheads and lines and the position of text in regard to these elements. At first single compounds are recognized and then eventually combined into a full dimension set by incrementally adding to the set.

To sum up this section, there are some reoccurring patterns. Firstly, we see that this topic has been researched for almost three decades without having a satisfying outcome yet, as all of the described systems and prototypes have great limitations or are highly specific. Another common factor is that vectorization was mainly used to extract graphical elements. We could also see that not only image-based drawings but also DXF files have been used as input files in multiple papers. A portion of the body of literature focuses completely on information tables or specifically the title block. Different kinds of documents are used, for example handwritten, printed, image or DXF format. If documents are scanned i.e. in image format an OCR technique has to be used. It is interesting that only one paper ([31]) mentions the use of deep learning algorithms as it is an increasingly used approach in a multitude of other domains. Especially for graphical recognition, neural networks (in particular convolutional neural networks) could prove to be useful. However, an application of machine learning in this context would require a wide range of annotated examples. [6] After this short summary it is fair to say that it is a complex, error-prone task, that is still relevant today. In the next chapter some general approaches for solving the stated problems will be discussed.

Chapter 3

Solution Design

To address the issues that are stated in the research questions, several approaches could be taken. The design and technology options will be discussed regarding each subproblem separately.

3.1 Use Case for Implementation

Automatic extraction of dimension requirements could be used for multiple use cases. However, for this thesis, a specific use case was the basis for the prototype. This use case is shortly described here and can be seen as the goal for this implementation. In a production setting, workpieces are produced as described above. After the production process itself, the workpiece is measured to check if it adheres to the conditions necessary for further use or the specifications in the contract. Measuring can be done manually or automatically with a measuring machine. For this specific use case, we assume that the measurement is done manually by an employee directly after the workpiece is produced. To know what has to be measured and what the measurements should be, the employee has to look at the engineering drawing. The measured dimensions are then noted to gather data that can be used to asses the overall production quality. The extraction process and user interface which are developed as part of this thesis should support this measurement process by providing the employee with a clear visualization of all relevant dimensions and tolerances as well as the applicable regulatory standards. In this it should serve as a component in a process driven quality control. Additionally, the user interface should allow the employee to enter the measured dimensions easily.

3.1.1 CAQ - Computer Aided Quality Control

As mentioned before, the prototype could serve as a component for process driven quality control. Automating the quality control of products, processes and services is also sometimes explicitly called computer aided quality control (short CAQ). CAQ includes all relevant aspects of quality control from planing to execution. [50] Therefore this prototype could also be seen as part of a computer aided quality control.

3.2 Extracting information from engineering drawings

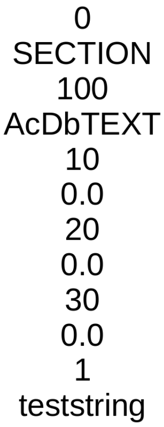
The system developed as part of the master thesis should be able to:

- extract all of the dimensional requirements
- including the nominal size values and respective tolerances
- as well as geometrical tolerances (form, position, run-out,..)

To achieve this result all values have to be extracted without losing information which values belong together. For geometrical tolerances, the symbols preceding these values are essential as well, so they have to be kept in the extraction process. The first step is to decide for a specific format of drawings. As mentioned in chapter 2 technical drawings can exist in multiple formats. Common formats are DXF, PDF, STEP or image formats (TIFF, PNG). These will be described briefly, focusing on the relevance for our research aims.

3.2.1 DXF

DXF stands for drawing interchange format, and was developed by AutoCAD. It is widely used in the industry, and therefore most CAD systems are able to work with DXF files. [5] Graphical as well as additional information can be included. DXF is written using ASCII symbols, therefore it is human as well as machine-readable, which makes this format a candidate for extracting information. The file consists of one column including group codes, that defines the datatype as well as the meaning of the following value. Subsequently, the value itself is noted. This can be seen in figure 3.1, where the value ”0” indicates the start of a new section, ”100” indicates that this section contains text, ”10”, ”20”, and ”30” represent the x, y and z-values of the text section and ”1” implies that the next value is the actual string that is contained in the drawing.There are existing tools (e.g. dxfgrabber [51])to extract information. Additionally as the format



```
0
SECTION
100
AcDbTEXT
10
0.0
20
0.0
30
0.0
1
teststring
```

Figure 3.1: Example of a DXF File.

is well documented [52] and explicitly structured, writing a new extractor should be feasible as well.

3.2.2 VDAFS

This standard only covers geometrical, no textual information. However, the successor "SET" does include tolerance information. It is, though, only used in specific industries and countries, therefore it will not be used in this thesis. [5]

3.2.3 STEP

"Standard for the Exchange of Product Model Data" or short "STEP" is the successor of the "IGES"-standard, which was also quite popular. "STEP" wants to include all information about a product and therefore serves as an all-in-one solution. [5] This would be ideal for extraction of information, or rather rendering the need for extraction as pointless. However, even though "STEP" as a format is commonly used for CAD models, it is not common to add information about tolerances (or non-graphical information in general). Additionally, not every version of "STEP" supports the addition of tolerances. [53]

The before mentioned standards are explicitly used for technical drawings or CAD models, which are not easily understandable for humans. However, often these drawings are also conveyed in PDF or TIFF, which convey the basic requirements at first glance. Additionally. most older drawings are mostly only available in one of these formats.

3.2.4 PDF

PDF files are commonly used for conveying pictures as well as text. However, PDF is a complex format, which was originally only intended to look the same on all platforms, not to be read automatically. Therefore, there are two different kinds. The first one are digital PDFs which can be read electronically, selected, searched, edited and also include meta information. The second kind of PDF documents are scanned or image-only PDFs, which can also include text, but everything is made up of one image, where searching or editing is impossible. [54] As the second kind is basically a picture format it belongs to the next category. If the drawing is in a digital PDF format, information can be extracted using different tools [55], either browser-based, command-line tools (e.g. pdftotext [56]) or libraries for different programming languages (e.g. PyPDF2 [57], tika[58] or textract[59] for python).

3.2.5 Picture Formats and OCR

If the technical drawings are in any image format (PNG, TIFF, JPG,..) the extraction process will be more difficult and error prone, as the values cannot be extracted directly, but OCR (optical character recognition) has to be performed first. To put it briefly, OCR applications try to isolate each letter and compare it to stored versions of letters to get the best match. This can involve different techniques including machine learning methods.[60] Even tough, this topic is very well researched, there is still no perfect system available, especially for unstructured documents [61]. Most of the related work (see section 2) use OCR as they are working with image formats. This is especially necessary if not only the textual components but also the graphical elements should be recognized. There are different proprietary as well as open-source OCR applications available (for python for example pytesseract, a wrapper for Tesseract, which was developed by Google [62]).

After having extracted the values and stored them in textual form, they should be recomposed to the get the nominal values with the respective tolerances as one group rather than having all values separately. To achieve this either regular expressions or clustering methods could be used.

3.2.6 Regular Expressions

As the composition of dimension sets (nominal size and tolerances, geometrical tolerances including symbol, value and additional information, see 2.6), follows specific patterns, regular expressions could possibly be used to recompose the individual values back to the dimension set, if the extraction was done in a way that preserves the order of the values. Regular expressions are a way to search for patterns in strings. These can be used in basically all programming languages by using libraries (for example the "re" module in python) [63].

3.2.7 Clustering

Another way to recompose the dimension set would be to apply clustering methods. Clustering methods are machine learning techniques to group similar objects (based on some similarity measure) together to create cohesive groups. There are different techniques like density or distribution-based clustering[64], each of these having implementations in different programming languages. The similarity measure can be a simple distance measure as well as more sophisticated measures.

3.2.8 Machine Learning

A completely different approach to tackle these issues are more complex machine learning techniques especially neural networks. Machine learning techniques could be used to extract the dimension sets without losing the connection between the nominal sizes and the tolerances. Especially convolution neural networks(CNNs) are systems that can be used for image recognition and classification. However, machine learning techniques are only applicable if a large base of training data is available. Training data means in this case annotated and labeled technical drawings. There are methods for artificial learning as well, which require less, but still some, training data. Annotating samples is a tedious and time-consuming task. Therefore, it is understandable, that machine learning has not yet been widely established in this field. Still, it could be a promising approach, especially for digitizing the drawing as a whole (including graphical elements). [6]

To summarize, we have different approaches for extracting text out of technical drawings. The first one consists of extracting the textual values and recomposing them to the respective dimension sets. The second approach being machine learning, where the whole dimension set could be extracted at once. Regarding the drawing format, DXF as well as PDF seems promising, as both include additional information and are easily accessible. These two formats also offer the benefit of having graphic and textual elements already separated. Image formats could be interesting as well, as most legacy drawings are stored as images. However, using OCR could lead to inaccurate results. In chapter 4 different approaches will be tried and the results are documented.

3.3 Integration of regulatory documents

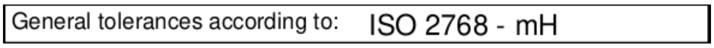
Another goal of this thesis is to evaluate how to integrate regulatory documents into a software system. As in Europe and in particular in Austria almost exclusively ISO standards (or the Austrian implementation of an ISO standard in the form of ÖNORM) are used, the focus lies on ISO regulations. However, in general an approach should be developed to include all kinds of additional regulatory material, for example ANSI standards or individual standards provided by a specific manufacturer. The requirements for this goal are to

- analyze which standards are relevant for a specific technical drawing
- extract the essential information out of these standards
- integrate this extracted data in the user interface
- provide a holistic overview over all relevant requirements for the workpiece

The following section will explore these requirements and describe some solution approaches.

3.3.1 Which regulatory documents are relevant?

If a drawing is meant to conform to ISO standards, all of its components have to meet the required ISO standards. This is necessary if technical drawings should be created in a standardized way and interpretable for everyone. ISO norms specify, as mentioned in 2, basically every aspect of technical drawings, like fonts or line widths. Even tough strict and thorough guidelines are important for standardization, they are not, however, for manually interpreting a drawing. Additionally, it would be overwhelming for the user to display all related standards. Therefore, it is important to select only essential regulatory documents, necessary for interpreting the drawing or measuring the workpiece, for integration into the prototype.



General tolerances according to: ISO 2768 - mH

FIGURE 3.2: Example of the Specification of the Tolerance Class.

There are different kinds of standards that could be relevant for a drawing. The first kind are the ones specifying the general tolerances. General tolerances, are the ones being applied to dimensions, where no explicit tolerance is stated. There are different

Tolerance Class		**Tolerances for Nominal Dimensions**						
Abbrevation	Name	from 0,5mm to 3mm	from 3 to 6	from 6 to 30	from 30 to 120	from 120 to 400	from 400 to 1000	above 1000mm
f	fine	±0,05	±0,05	±0,1	±0,15	±0,2	±0,3	±0,5
m	medium	±0,1	±0,1	±0,2	±0,3	±0,5	±0,8	±1,2
c	rough	±0,2	±0,3	±0,5	±0,8	±1,2	±2	±3
v	very rough	-	±0,5	±1	±1,5	±2,5	±4	±6
For nominal dimensions under 0.5mm the tolerances should be noted directly at the nominal dimensions.								

Table 3.1: Example of a Table Specifying General Tolerances.

tolerance classes. The combination of tolerance class with the actual dimension specifies the tolerances to be applied to the dimension. The tolerance class should be mentioned in the drawing, see for example figure 3.2. ISO 2768 (part 1 and 2) is the document which specifies the general tolerances and ”m” (found in ISO 2768-1, specifying measures for linear and angular dimensions and bevel length) as well as ”H” (found in ISO 2768-2, specifying tolerances for geometric measures as for example perpendicularity and straightness) are the applicable tolerance classes [65]. These tolerance classes tell us where to look in the tables, one of these being displayed in table 3.1 [65]. To get the applicable general tolerances we have to combine the tolerance class, e.g. ”m” with the value of the nominal dimension, to get the acceptable value of deviations.

The other kind of relevant documents are the ones specifying not the actual size of the tolerances but rather the definitions of the tolerances, for example the exact meaning of the geometrical tolerance symbols, see figure 2.6 or figure 2.7. ISO 1101, is the ISO document, which lists all applicable symbols and defines them. This standard is valid for all drawings, however it could also be explicitly stated, like in figure 2.3. Another example would be the standard regarding the surface quality, namely ISO 4287.

For an experienced employee it may not always be necessary to look up either kind of standard, however in special cases or just to ensure correct interpretation it is useful to have the relevant standards in close reach, or in this case integrated in the prototype.

3.3.2 How to extract information?

As mentioned above the information in these standards can exist either in unstructured text form or in tabular form. To extract information, different extraction methods may be useful. To extract information from the textual parts, a simple search may be enough, as mostly definitions are in text form. A more sophisticated and exploratory approach using text mining methods, which extracts all information regarding a specific definition, could also be possible, see for example other papers dealing with extraction of information from regulatory documents [66].

However, most of the relevant information is locked in tables, which are more difficult to extract. The first issue is that ISO standards are in PDF formats, and as mentioned before, PDF can be quite complex to extract out of. Especially for extracting knowledge out of the tables, the PDF document would have to be transformed to text format first, and then the table would have to be recomposed, either by using a tool specialized in extracting tables from PDF (for example [67]) or by extracting the text via a command-line tool or library and then recomposing the table by using coordinates or heuristics.

Another issue is, that we need to define terms that are important and can be found in the relevant ISO standards, as well as in which ISO documents these can be found, otherwise all documents have to be searched, which would be inefficient. To tackle this issue an ontology could be developed. An ontology can store knowledge about a domain in a structured form, and is often used for knowledge extraction out of unstructured documents(for example [68] or [69]). However in this case the ontology could be used to store information about which ISO documents hold which information or even more straightforward which general tolerances are applicable for which nominal values and tolerance classes. It would take considerable effort to create these ontologies, as all of the values would have to be extracted manually or a script would have to be written.

The here mentioned approaches will be further elaborated in chapter 4.

3.3.3 How to include the information?

Related to the issue of extracting information out of ISO documents is the question of how to display the extracted information in a useful way. It may not be necessary to include all related information in the user interface as these are not needed all the time. However, it may be useful to include links to all applicable standards, so that employees can access them easily. Additionally, parts that seem relevant could be highlighted or a search function could be implemented that the employee can jump around relevant parts of the ISO document. If applicable, general tolerances can be extracted from the documents or if they are stored in an ontology, they can be displayed next to the nominal dimension in the user interface.

3.4 Generation of User Interface

Another aim of this thesis is to create a user interface that could be used in production processes for supporting measurement of the workpiece or just as a general overview of the workpiece specifications. This user interface should meet the following requirements:

- display the drawing
- display all extracted dimensions and tolerances
- include additional information (see previous section)
- provide input fields for actual measurements
- provide input fields to specify if the respective dimension is relevant to the overall validity of the workpiece

Additionally, all of the information should be presented in a way that it is easily understandable. The extracted measurements should have references to the exact position in the drawing, so that employees can see at first glance where the measurements have to be taken. Another requirement is that the user interface can be generated automatically for a new drawing and no adjustments have to be made.

3.4.1 Interaction Design

The user should be able to upload a drawing, then the extraction process can take place in the background. As soon as the extraction is finished, the drawing as well as the extracted dimensions are displayed. The user can then click on each dimension to see it highlighted in the drawing. If the workpiece is measured, the user can input the actual size next to the extracted dimension. Additionally, the user can supply the system with information if the dimension is necessary for the correct functioning and usage of the workpiece. These inputs are stored in a database and can be further used for data analysis or adaption of the system.

3.4.2 Technologies

For the user interface HTML as well as ”javascript” can be used, as these technologies are basically standard for web interfaces. Moreover, we have to provide ways for client and server side to interact for example via AJAX-calls. The goal is to extract and provide all necessary data in the extraction process so that the web application only needs to fill in the variables to create the user interface. The interaction between the web application and the extraction process should be well defined, so easy adaption and reuse can be ensured.

Chapter 4

Implementation

In section 3 multiple approaches to solve the before stated research questions were covered. The developed prototype is described in this chapter, including a description which approaches have been taken.

A goal of the implementation was to develop several modules that could be used on their own or in conjunction. The user interface and the extraction process are completely independent, so each can be reused or adapted easily. The whole application is written in Python 3.7. Python was used as the implementation language, as it is a widely used programming language, especially for data mining. There is a multitude of existing libraries for all kinds of purposes and python scripts are easy understandable even for people who have little knowledge about programming. Additionally editing and adapting is fast and simple as well.

The drawings used for developing and testing the prototype are all used at the ACDP, however not all of them can be displayed in this thesis, as some drawings are copyright protected by partner companies.

4.1 Reading Measurements from Engineering Drawing

For the information extraction, section 3 mentioned the possibility of using neural networks. However it was not used for this prototype, as firstly, not enough sample drawings were at disposal and secondly control over what is important and should be extracted was desired. The other mentioned approaches are using different drawing formats to gain information. The formats that were tried out in the process of developing the prototype are DXF, image formats and PDF.

4.1.1 DXF

As mentioned in chapter 2, DXF files are composed of ASCII symbols and have a clear structure. So reading information out of DXF files should be fairly easy. The first attempt included using a library called dxfgrabber [51], which should be able to read DXF files. However, the whole file is read and all the information about the drawing is included. It is therefore more complex than what is needed for this prototype. A search concluded that no other libraries exist that specialize just on reading textual information out of a DXF drawing. For this reason, a short python script was written. The script reads each line in a buffer, if the line contains the number "100" we know that a new textual element starts here. Everything that was written into the buffer before is then fed into another function and the buffer is emptied, ready to search the remaining lines. The second function converts the content of the buffer into a dictionary, alternating between keys and values, meaning that the first row is a key, the second the respective value and so on. This follows the DXF logic. We can then look for the key "1", which is the textual value itself, as well as the keys "10" and "20", which are x and y-coordinates. These three values are then written to a CSV file, which is then used for further refinement.

This script is fairly short and easy, but is able to extract all textual information out of the drawing. However, there is more textual information than only dimension sets. Furthermore, each value was extracted separately, which means the dimension sets are split up. Therefore more steps have to be taken to get the recomposed dimension sets. The separation of relevant and not-relevant information, could possibly be done by regular expression and the recomposition of groups could be done using clustering, as the coordinates are extracted as well, and give hints about which values belong together. The coordinates are relative coordinates, this means they are in relation to a specific object. From first inspection it was not clear, to which objects they are relative to.

While in the process of analyzing this issue, another issue became obvious. Not enough DXF drawings were at disposal. For this reason, the implementation of a DXF reader was paused, and the focus was shifted on the other formats. However, extraction of the dimensions should, in principle, be possible. More work on recomposing the dimensions and tolerances would have to be done.

4.1.2 Image formats

As most legacy drawings are available in PNG, TIFF, or as a scanned PDF, image formats are particularly interesting. As mentioned in section 3 OCR has to be applied to images, before starting to analyze the content. There exist multiple OCR libraries for python. For this prototype the library tesseract [62] was used. However, the accuracy was very low, not even half of the dimensions were recognized, which might be caused by interference of the graphical objects. Additionally, if the drawings are in PDF format, they has to be converted to a "true" image format like TIFF. Because of the poor result and the tedious process, the next approach was tried.

4.1.3 PDF

As the sample drawings are all in digital PDF (which means searchable), this approach did not need to include OCR, but is able to read directly from the PDF file. There are different python libraries for reading PDF like PyPDF2 [57], textract[70] or Tika [58]. Trying these resulted only in poor extraction outcomes, especially because the dimension sets were completely decomposed. However, there also exist command-line tools for PDF extraction like the the xpdf-tools [56]. These tools include, among others, a "pdftotext" tool, which converts PDF to ".txt" format and "pdftohtml", which converts the file to an HTML file.

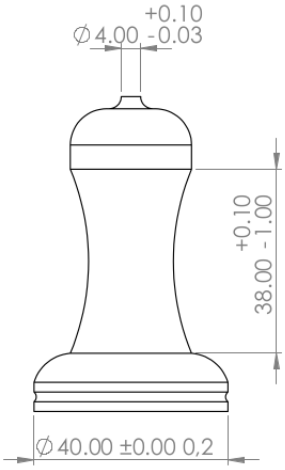
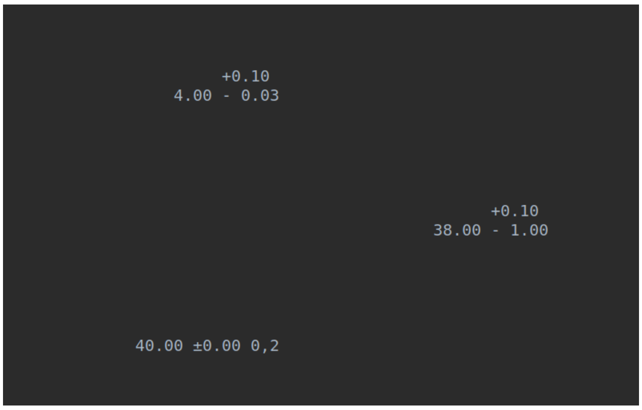


Figure 4.1: Sample Drawing Used for Analyzing Different Approaches.

The initial attempt consisted of using the "pdftotext" tool including the option "-layout" which tries to maintain the original layout. The following command:

```
pdftotext -layout path_to/sample.pdf
```

leads to this text result (see figure 4.2) for a simple drawing, see figure 4.1. As can be seen, all dimension sets have been extracted and kept spatially near in the text file.



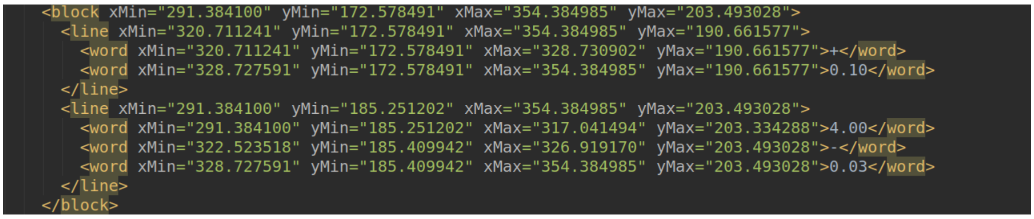
```
          +0.10
      4.00 - 0.03


                                     +0.10
                                 38.00 - 1.00



  40.00 ±0.00 0,2
```

Figure 4.2: Extracted Dimensions.

To extract these sets from the text file, the white space between the characters could be used as a distance measure. However, it is even easier if the option ”-bbox” is used. The following command, extracts all textual information from the PDF, including the coordinates of the bounding box of each value, in an HTML file.

```
pdftotext -bbox-layout path_to/sample.pdf
```

Figure 4.3 shows an extract of this HTML file. In most cases, one dimension set is extracted into one block. Each value being one word. The values are not always in the correct order, but with the coordinates they can be recomposed into the correct order.



```
<block xMin="291.384100" yMin="172.578491" xMax="354.384985" yMax="203.493028">
  <line xMin="320.711241" yMin="172.578491" xMax="354.384985" yMax="190.661577">
    <word xMin="320.711241" yMin="172.578491" xMax="328.730902" yMax="190.661577">+</word>
    <word xMin="328.727591" yMin="172.578491" xMax="354.384985" yMax="190.661577">0.10</word>
  </line>
  <line xMin="291.384100" yMin="185.251202" xMax="354.384985" yMax="203.493028">
    <word xMin="291.384100" yMin="185.251202" xMax="317.041494" yMax="203.334288">4.00</word>
    <word xMin="322.523518" yMin="185.409942" xMax="326.919170" yMax="203.493028">-</word>
    <word xMin="328.727591" yMin="185.409942" xMax="354.384985" yMax="203.493028">0.03</word>
  </line>
</block>
```

Figure 4.3: Extract of HTML File.

The next step is to extract the values, as well as the respective coordinates from the HTML file using ”BeautifulSoup”, a python library for extracting data out of HTML files. Everything is then stored in an array of arrays. Each sub-array contains the value and the respective minimal and maximal x and y-coordinates from one block. If the block is unordered, the words are ordered within a block by x-coordinates using the built-in ”sort” function. To determine if a block is unordered a new function was written that compares the x-coordinates of all words in a block. If there are x-coordinates that are higher in the first words than in the later words we assume that it is not in order. For the simple drawing, this was enough to get all dimension sets in the right order and composition. However, for more complex drawings, just ordering them is not enough, as not all dimension sets are extracted as one HTML-block. Additionally even for a simple drawing, not only the dimension sets are extracted at this point, but all textual values, so further refinement is necessary. Furthermore, the symbols are not extracted correctly, for example the symbol for perpendicularity is replaced by the letter "f". Fortunately, the symbols are always replaced by the same letter. For these issues more post-processing work is needed. Two solve the problem of recomposing the dimension sets two approaches are possible: regular expressions or clustering.

4.1.4 REGEX

In any case, regular expressions are used directly after all of the values are read in from the HTML file, to extract the textual information about regulations. The first thing that is extracted are all textual values containing the words "ISO" and "EN" to get the explicitly mentioned regulatory standards. All of the found regulations are collected and stored in an array. Only "ISO" and "EN" are searched for, as these are the only ones occurring in the sample drawings. However, the regex could be easily expanded to include "ÖNORM" or "ANSI" as well. Additionally regular expressions are also used to clean the information of everything that is not related to dimensions or regulations. This will be explained in more detail later on.

As mentioned before, regular expressions could also be used to recompose the dimensions and tolerances by using patterns. Most dimension sets consists of the pattern "nominal dimension, +, upper tolerance, -, lower tolerance". Therefore it would be possible to look at value by value, comparing it to the values before and after that and recomposing a dimension set, based on whats missing to create a full dimension set. This works for some drawings and some dimensions. However, the pattern can vary quite widely, as it is possible to have no tolerances at all, two positive tolerances, only one tolerance, and more combinations. Therefore there are many patterns that should be taken into consideration. Even if more patterns are represented as regular expressions and searched for it, it only works if the values that belong together are in next to each other in the array, as coordinates are not regarded. In conclusion this method does work for simple drawings, however, it is not working for more complex samples, and is overall not an elegant method, as it is only based on heuristics.

As regular expressions did not work out, the next approach is to use clustering.

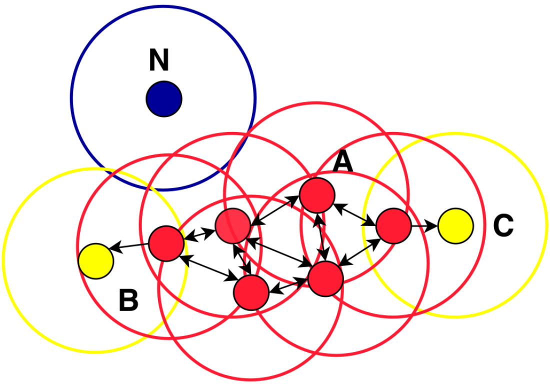


Figure 4.4: Clustering Using DBSCAN

4.1.5 Clustering

Clustering methods are unsupervised machine learning techniques, see section 3. There are different clustering techniques, all useful for different cases. For this case, we had to use a clustering method, that does not need to know beforehand how many clusters there will be, as we simply do not know. DBSCAN is one method where this specification is not needed. Additionally, we can specify the minimum amount of points necessary and the maximum distance between points to form a cluster. [71] Therefore DBSCAN was the optimal technique. DBSCAN implementations are available for most programming languages. The basic principle is that the user, sets the parameter "minPts", which defines how many points should at least be in a cluster and "ϵ", which defines the radius for each cluster. The algorithm looks at all points and groups the ones together that are within the specified radius. Figure 4.4 [71] shows an example of this algorithm. "N" is a noise point, not belonging to any cluster, whereas "A" is a core point, as it has "minPts" in its radius. The points "C" and "B" are border points, as they are in radius of a core point, but have less than "minPts" in their radius.

For this prototype the array from the extraction process, including values and coordinates, is used in conjunction with the "sklearn" clustering library [72], which includes a DBSCAN implementation. But before the clustering algorithm can be applied some pre-processing has to be done. As mentioned before, the array stores sub-arrays of the HTML blocks, which further include words and the respective coordinates. For the clustering process only the outer coordinates (the bounding box) is needed. Therefore a function was written, which extracts the maximum and minimum x and y-coordinates for each block. Additionally we try to analyze if the block is horizontal or vertical for later purposes. This is done by looking at the ratio of x to y coordinates. If x maximum and the x minimum are further apart than y maximum to y minimum, it is assumable that the bounding box is horizontal. The result is then stored as well. Before the clustering algorithm can be applied, a distance metric has to be defined. There are predefined metrics available, however, as we are working with bounding boxes, not points, a simple metric is not enough as it is not defined which distance exactly is measured. The measure used for this thesis, is the distance between the two nearest corners of two rectangles. A simple function was written that compares the distance of all four respective points of two rectangles, finding the closest points, and using this distance as the distance metric. If the rectangles are intersecting, the distance is set to 0. The distances are calculated for all bounding boxes and stored in a matrix. This matrix is then used as input for the clustering algorithm.

```
db = DBSCAN(eps=eps, min_samples=1, metric="precomputed").fit(dm)
```

The labels (which element belongs to which cluster) are then together with the inital element stored in a dataframe. This dataframe is then sorted in a way that all elements from one cluster are combined into one element. For the sample drawings this lead to a good accuracy, see section 5.

The parameters that can be set in DBSCAN are, as mentioned before, the minimal points in the cluster (here "min_samples"), and the *epsilon* value (here EPS). "Min_samples" is set to 1, as it is enough to have one point in the cluster to be a cluster. The setting of the optimal "ϵ" value is more complex. First, the value was adjusted so that the first sample drawing showed optimal result by manually trying different values. As more sample drawings were analyzed, is became obvious that this value is not optimal for all drawings. Therefore the value is set dynamically for each drawing by examining the amount of words and blocks. The more of these exist, the more complex is the drawing and the more the values have to be clustered together. At the moment the EPS value is set at 7 if there are more than 500 words and at 1 for less words. However, this heuristic can be adjusted and refined as more drawings are analyzed.

After the clustering the resulting data frame is then converted into a dictionary where the dimension set is the key and the coordinates (x-min, x-max, y-min, y-max), stored in an array, are the respective values. This makes it easier to work with the elements for further refinements, as the arrays got big and difficult to work with.

The next step is to search if a title block and additional tables are present in the drawing. This is again done by looking for specific keywords, that usually belong in such tables (e.g. for the title block the search terms "start drawing" are used, as most sample drawings used for this thesis include a list of all revisions). These keywords can and have to be adjusted manually if drawings, that do not adhere to these heuristics, are read in. At this point we know if and if yes, how many tables are present in the drawing.

After this step is done, the textual information is cleaned using regular expressions, to just include relevant information. For example all words are excluded (as we already extracted information about the regulatory standards before), as well as individual letters, which only mark the drawing area on the paper. Additionally, the symbols for geometrical tolerances are not extracted properly, so this data cleaning stage includes converting the letters, which are representatives of the symbols, back to the unicode symbols by using if-else statements.

The last step in the extraction process is to organize the dimensions by the views they belong to. Views show different sections of the drawing in more detail. See for example figure 4.5 and figure 4.6.

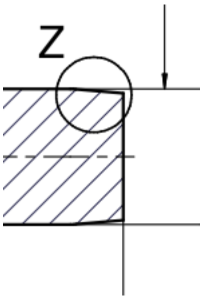


Figure 4.5: Part of Workpiece Which is Shown in a Detail View.

Figure 4.5 shows how it can be indicated that a detail view is available. The "Z" is the label of the section. The respective section is displayed in figure 4.6, including the scale of the specific view. To create a clearly arranged user interface the values should be organized by the view they belong to. This can also be used to organize the values by importance, as it is common to name the views in alphabetical order, starting with the first and most important view at "A" and continuing in the order of importance.

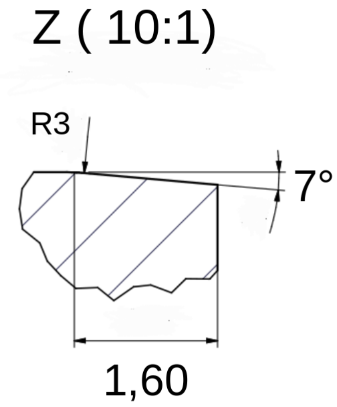


Figure 4.6: Detail View of Drawing.

The organization of values in views is easy to grasp for a human being as we identify on first sight which elements belong together. However, it is complex to achieve this automatically, as there is no clear connection between the values, especially because we solely focus on textual elements without including the graphical elements. The first round of clustering is only done to cluster on a very basic level (namely the dimension sets), therefore one approach could be to apply a second round of clustering. Having said this, after reviewing some sample drawings, it is often the case that values are spatially closer to another view than to the view they belong to. This complicates the clustering process as we cannot simply use spatial distance metrics.

Another approach is to use heuristics. In most drawings the views are organized by title of the view and the scale (see figure 4.6). The approach is to find all titles and from there on check where the next section to the left and right and the next section above and below are and partition the drawing in rectangles confined by the neighboring views. The first step is to use regular expressions to get the different views. The conventions for naming the views are different, therefore it can be necessary to manually adapt the regular expressions. The tables that have been extracted in a step before are also taken into account as a view. The found views are stored in an array including the coordinates of the view title. After that, the array is sorted by the x-coordinate, so we have the detail view that is closest to the left side (origin of the x-coordinate). The next step is to get the rectangles. To achieve this, each view is compared to all other views and the neighboring views (regarding x and y direction) are detected by finding the shortest distances. If no neighboring views can be found in some direction (e.g. below) we assume that there is no views below it and the edge of the drawing is considered the confining border of this view. To get the border to the neighboring views the distance between the two views is divided in half and at this point the border is set. The resulting x and y values are then stored in an array. To match all of the values to their respective views, a function checks for each value, namely the rectangle that is built from the value coordinates, if it intersects with the view rectangle defined before. If they intersect it is assumed that the value belongs to this view. All values and views are stored in a dictionary where the view title is the key, the element is the value and also serves as a key for getting the coordinates of the value. The coordinates of the view rectangle is also stored in the dictionary to be used later in the user interface. This approach is, of course, only a heuristic approach and will not work on every drawing, but it did work for sample drawings and can be adjusted for other drawings as well. The last step in the extraction implementation is to store all the extracted elements in "redis" [73], which is a key-value store. The dictionaries are first converted into json-format and then stored in "redis".

In this section different approaches have been described that could be used, and which approaches are used for this prototype. In principle, different extraction approaches could be used to achieve the same results. The DXF approach would need some post-processing to find the absolute coordinates, but could then still be used as basis for clustering. The OCR approach could be used as well, however the results would be less accurate.

Some questions that remain are: What are the most important and most relevant dimensions in a drawing? Is there a pattern that for certain workpieces, some dimensions are more important? Additionally, without the graphical elements there is no way to recognize which dimension is a basic dimension (theoretically accurate measure) and which is a nominal dimension without explicit tolerances. These issues can be approached in future work.

The next section covers the extraction of additional information from regulatory standards.

4.2 Getting Additional Information from Regulatory Documents

The second research question deals with additional regulations regarding dimensioning and tolerances. For this thesis, only ISO and DIN documents are taken into consideration. These are either mentioned explicitly in the drawing or are generally applicable for a specific subject (e.g. surfaces). To include this additional information there are different approaches as mentioned in section 3. At this point an interview with a domain expert was conducted to figure out which parts of the ISO regulations are the most relevant and which are needed to interpret the drawing correctly. The domain expert explained that people working regularly with these drawings rarely need to check the standards, and if they do mostly the tables are relevant. Firstly, the general tolerances and the respective tolerance class can be checked. Secondly details about fits and threads (which are commonly labeled as M8, H7,..) with the label indicating the specific category, can be quite important. Therefore the first approach was to convert all available ISO document from PDF to text to further extract and condense information in particular from the tables. The thereby obtained information could either be used to build an ontology or to directly note the related standard and value in the user interface next to the dimension requirement.

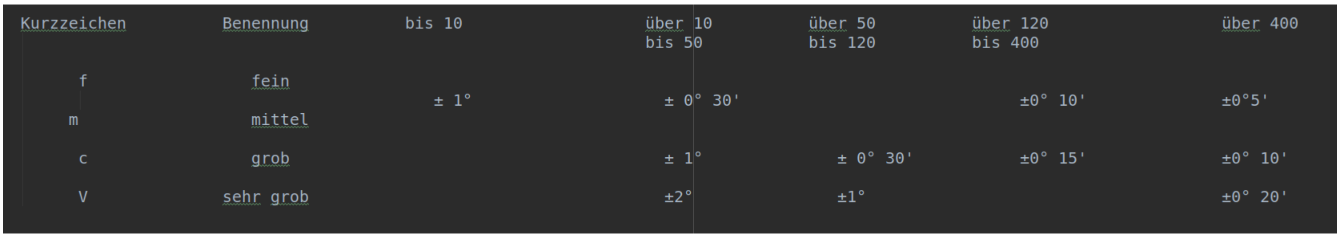
4.2.1 Table Extraction

As "pdftotext" worked for the technical drawings, it was assumed that it could also work for ISO documents. A test confirmed that it works equally well for the purely textual information. The tables were extracted as well, including most of the format. See table 4.1 [65] for the original table and figure 4.7 for the respective extraction. However, as can be seen, some values are missing.

Toleranzklasse		**Grenzabmaße für Längenbereiche in mm, für den kürzeren Schenkel des betreffenden Winkels**				
Kurzzeichen	Bennenung	bis 10mm	von 10 bis 50	von 50 bis 120	von 120 bis 400	über 400mm
f	fein	±1°	±0°30’	±0°20’	±0°10’	±0°5’
m	mittel					
c	grob	±1°30’	±1°	±0°30’	±0°15’	±0°10’
v	sehr grob	±3°	±2°	±1°	±0°30’	±0°20’

Table 4.1: Example of Tolerance Specification in ISO2768-1

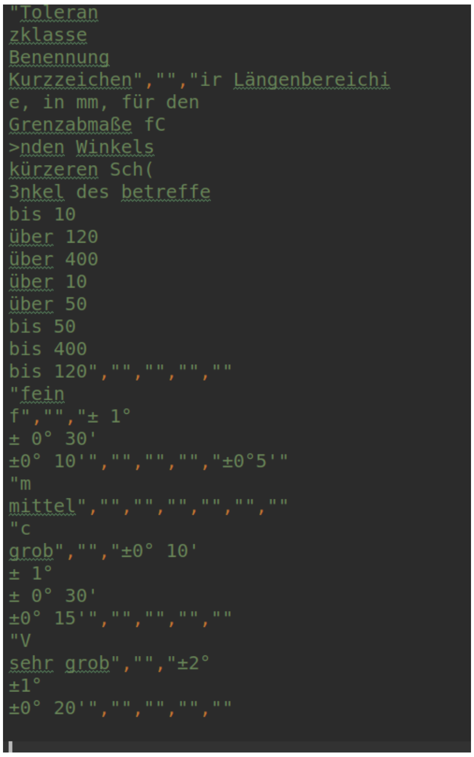
As it is in text format, a script would be needed to transform it back into a table. This could either be done by looking at the rows and white spaces in between characters or by using coordinates, similar to the extraction technique from chapter 4.1.



```
Kurzzeichen          Benennung          bis 10                   über 10          über 50          über 120                  über 400
                                                                 bis 50           bis 120          bis 400

      f                 fein
                                           ± 1°                    ± 0° 30'                             ±0° 10'              ±0°5'
     m                  mittel

      c                 grob                                       ± 1°              ± 0° 30'           ±0° 15'              ±0° 10'

      V              sehr grob                                     ±2°               ±1°                                     ±0° 20'
```

FIGURE 4.7: Table Extracted Using "pdftotext".

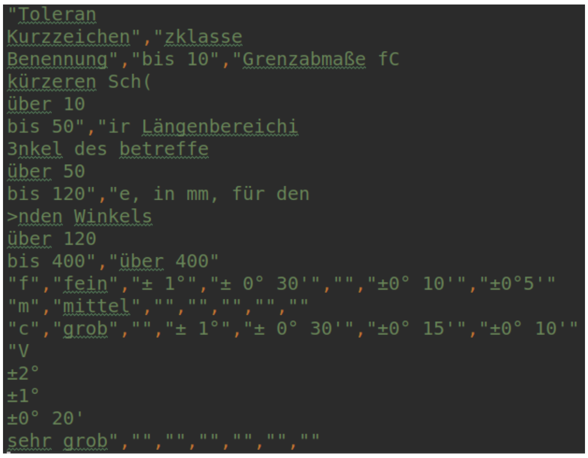
Before continuing with this approach, other, specialized tools were used to extract only the tables from the PDF. Firstly the "Camelot"[67] implementation was used. It is a tool that is specialized on extracting tables out of PDF files and convert it to either JSON or CSV. However, the first try did not achieve a clean extraction as can be seen in table 4.8. The layout was not maintained and it would be more complicated to recompose compared to the "pdftotext" solution.



```
"Toleran
zklasse
Benennung
Kurzzeichen","","ir Längenbereichi
e, in mm, für den
Grenzabmaße fC
>nden Winkels
kürzeren Sch(
3nkel des betreffe
bis 10
über 120
über 400
über 10
über 50
bis 50
bis 400
bis 120","","","",""
"fein
f","","± 1°
± 0° 30'
±0° 10'","","","","±0°5'"
"m
mittel","","","","","",""
"c
grob","","±0° 10'
± 1°
± 0° 30'
±0° 15'","","","",""
"V
sehr grob","","±2°
±1°
±0° 20'","","","",""
```

FIGURE 4.8: Table Extracted Using Camelot

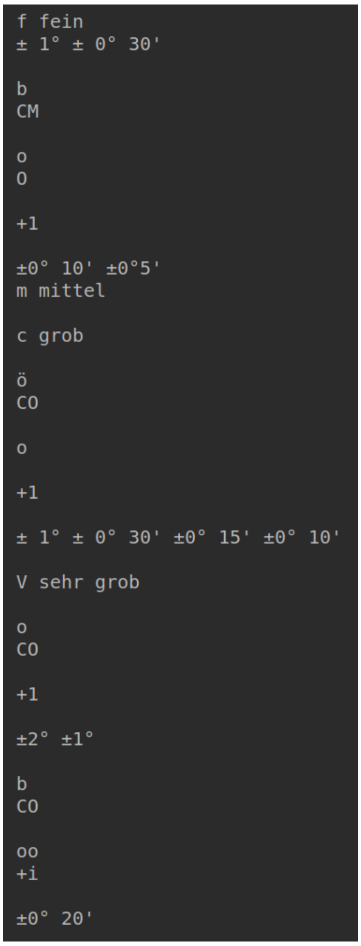
The "camelot" library allows to adapt certain parameters to better fit the needs of particular tables. After trying out different parameters the best fit is shown in figure 4.9. It is able to identify most of the layout and the values that belong together but still not all rows and values were identified correctly. In this example the last row cannot was identified as a row.



```
"Toleran
Kurzzeichen","zklasse
Benennung","bis 10","Grenzabmaße fC
kürzeren Sch(
über 10
bis 50","ir Längenbereichi
3nkel des betreffe
über 50
bis 120","e, in mm, für den
>nden Winkels
über 120
bis 400","über 400"
"f","fein","± 1°","± 0° 30'","","±0° 10'","±0°5'"
"m","mittel","","","","",""
"c","grob","","± 1°","± 0° 30'","±0° 15'","±0° 10'"
"V
±2°
±1°
±0° 20'
sehr grob","","","","","",""
```

Figure 4.9: Table Extracted Using Adapted "camelot" Script.

Another python tool that can be used for extracting tables out of PDF documents is "tika" [58], which is actually an "Apache" tool, but has a python implementation. The result can be seen in figure 4.10. Similar to the "camelot" tool it was not possible to extract the layout correctly.



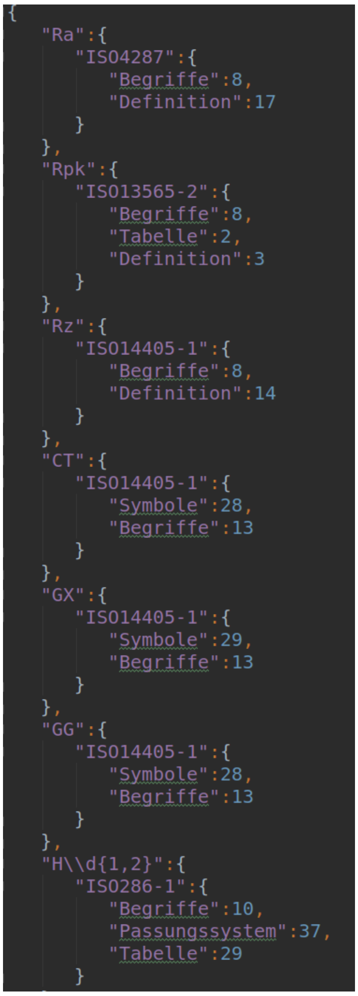
```
f fein
± 1° ± 0° 30'

b
CM

o
O

+1

±0° 10' ±0°5'
m mittel

c grob

ö
CO

o

+1

± 1° ± 0° 30' ±0° 15' ±0° 10'

V sehr grob

o
CO

+1

±2° ±1°

b
CO

oo
+i

±0° 20'
```

FIGURE 4.10: Table Extracted Using ”tika”.

As can be seen, the extraction of tables is a complex task, leading to sub-optimal solutions. A less complex but tedious approach would be to manually extract important values and build an ontology which is then used to determine the respective regulatory document and value. Another approach, which will be further pursued due to its simplicity and elegance, is to build an ontology where only to the relevant document is stated, the document is then linked in the user interface and particular useful search terms are included as well. This approach allows the user to directly open relevant regulatory documents, to look up a value and to jump around the essential terms within the document using the predefined search terms.

4.2.2 Ontology

The ontology should contain key terms contained in the drawing that relate to specific regulatory documents. For example, as mentioned before, the term ”H7” stands for a fit, which is defined in ISO-286. The ontology is created manually and can be adjusted to new drawings which mention regulatory standards that have not been included so far. As for this thesis only a limited number of standards are relevant and there are no complex interconnections between the terms and the standards, a simple JSON file (called ”config.json”) is used to depict these relations. In future work an ontology using traditional ontology-tools and languages like RDF and OWL could be developed.



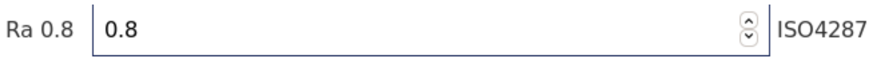
```
{
   "Ra":{
      "ISO4287":{
         "Begriffe":8,
         "Definition":17
      }
   },
   "Rpk":{
      "ISO13565-2":{
         "Begriffe":8,
         "Tabelle":2,
         "Definition":3
      }
   },
   "Rz":{
      "ISO14405-1":{
         "Begriffe":8,
         "Definition":14
      }
   },
   "CT":{
      "ISO14405-1":{
         "Symbole":28,
         "Begriffe":13
      }
   },
   "GX":{
      "ISO14405-1":{
         "Symbole":29,
         "Begriffe":13
      }
   },
   "GG":{
      "ISO14405-1":{
         "Symbole":28,
         "Begriffe":13
      }
   },
   "H\\d{1,2}":{
      "ISO286-1":{
         "Begriffe":10,
         "Passungssystem":37,
         "Tabelle":29
      }
```

Figure 4.11: Example of a Configuration File.

Figure 4.11 shows an example. The file is constructed as a dictionary of dictionaries with symbols and keywords as keys for the regulatory documents, the names of the regulatory documents as keys for the search terms. In this example the search terms are general terms like ”definition”, ”terms” and ”symbols” which refer to specific sections in the ISO standard. The keywords are all formulated to be used as regular expressions, so either the terms itself, the unicode expression (for the symbols) or a regular expression (here for fits and threads definitions) are used. This file is read in and all of the in the extracted textual elements are compared to the file contents. If a match is found, the name of the regulatory document and the search terms are stored and subsequently linked in the user interface.

4.2.3 Linking and Searching ISO Documents

The second part of the pursued approach is to link the regulatory standards in the user interface as well as making them searchable using the specific keywords. The regulatory documents have to be provided by the user and uploaded in the folder ”static/isos”. Only the provided standards can be linked. The linking process starts by comparing all extracted information from the drawing with the terms stored in the ”config”-file to check for relevant terms and their respective documents which was described with the example above. The next step is looking up in the file system if this document is available. If this is the case, the document is directly linked in the user interface next to the key term as can be seen in figure 4.12.



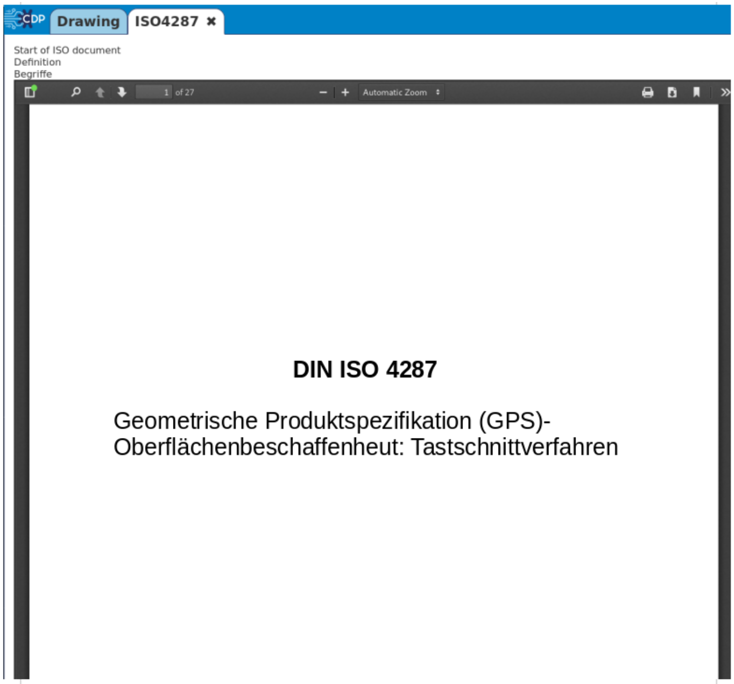
Ra 0.8

0.8

ISO4287

Figure 4.12: Linking the ISO Document Specifying Surface Norms Next to the Key Term ”Ra”.

Otherwise, the name of the document is noted in the UI, with the additional remark that it is not available. In addition, if there are general applicable regulatory documents mentioned in the textual parts of the drawing, these are linked as well on top of the user interface, see figure 4.15. This will be described in more detail, including sample pictures, in the section 4.3. If the document is available the user can click on it and a new tab loads the file. The specified search terms are noted above the file, so the user can click on them to jump to the specific sections in the document. Additionally, there is a button to go back to the start of the document. This can be seen in figure 4.13. This is done by using the default ”pdf.js” implementation, that is used in Firefox as the default tool for displaying PDFs. Using this tool, we can skip to specific pages in a pdf by document by specifying the page in the url (for example: ”centurio.work/edi/show/GV_12#pdfpage=3”).



Start of ISO document
Definition
Begriffe

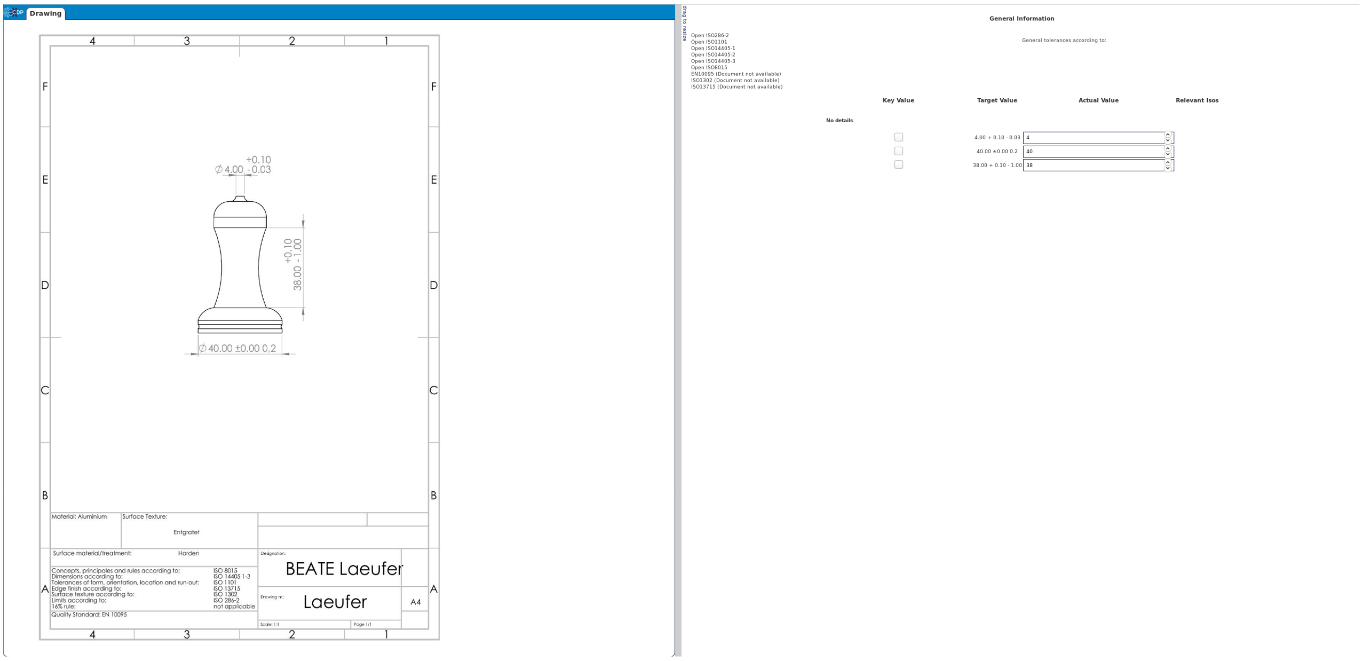
DIN ISO 4287

Geometrische Produktspezifikation (GPS)-
Oberflächenbeschaffenheut: Tastschnittverfahren

Figure 4.13: A New Tab Showing the ISO Document and the Respective Search Terms.

4.3 User Interface

The user interface combines the previously described parts and should create a concise overview over the drawing and the dimension requirements as well as providing input possibilities for measurements. The user interface was written in python using the ”flask” web framework [74]. The implementation of the web application is independent of the extraction implementation. The two programs interact mainly using ”redis”. Figure 4.14 shows an overview of the user interface, zoomed out to see the entire drawing. More details will be described in the following sections.



Material: Aluminium

Surface Texture: Entgratet

Surface material/treatment: Harden

Concepts, principales and rules according to:	ISO 8015
Dimensions according to:	ISO 14405 1-3
Tolerances of form, orientation, location and run-out:	ISO 1101
Edge finish according to:	ISO 13715
Surface texture according to:	ISO 1302
Limits according to:	ISO 286-2
16% rule:	not applicable

Quality Standard: EN 10095

Designation: BEATE Laeufer

Drawing nr.: Laeufer

A4

Scale: 1:1

Page 1/1

General Information

Open ISO286-2
Open ISO1101
Open ISO14405-1
Open ISO14405-2
Open ISO14405-3
Open ISO8015
EN10095 (Document not available)
ISO1302 (Document not available)
ISO13715 (Document not available)

General tolerances according to:

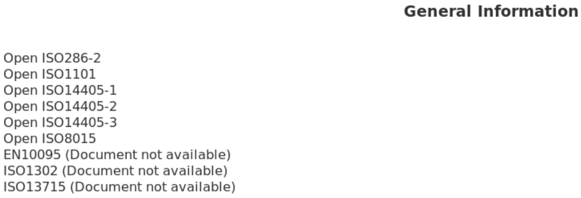
No details

Key Value	Target Value	Actual Value	Relevant Isos
□	4.00 + 0.10 - 0.03	4	
□	40.00 ±0.00 0.2	40	
□	38.00 + 0.10 - 1.00	38	

FIGURE 4.14: Overview of Whole User Interface.

4.3.1 Interaction Design

The goal of the web application is to support the measuring process in the production. For this we want to display the drawing itself as well as the extracted information. The user interface includes two columns which can be adapted by a bar in the middle of the screen that can be dragged to resize the respective windows. The left side includes the drawing, whereas the right side holds all of the information. The upper part is called ”General Information” and displays all general regulatory standards that have been extracted from the drawing as well as general dimensional requirements, if the drawing did contain them, for example, in the title block. If the regulatory documents are available on the server, they are linked. Otherwise they are marked as ”not available”. This block can be seen in figure 4.15



General Information

Open ISO286-2
Open ISO1101
Open ISO14405-1
Open ISO14405-2
Open ISO14405-3
Open ISO8015
EN10095 (Document not available)
ISO1302 (Document not available)
ISO13715 (Document not available)

FIGURE 4.15: Additional Regulatory Documents.

Below this block, the actual measurements are noted. This block is partitioned into four columns. The first one is called ”Key Value” and contains checkboxes, where a user, or potentially the ”first” user measuring a workpiece, can check if a value is essential to the functionality of the workpiece. The second column is called ”Target Value” and contains the actual extracted values (all dimensions including tolerances). Right next to it is the column ”Actual Value”, the input field where the user can enter the measured value. The last column ”Relevant ISOs” contains a link to applicable regulatory documents, if any exist and if the documents are available. This block can be further sectioned by the different views. The views are ordered alphabetically, as we assume that this relates to the importance of the views. If the drawing did not contain multiple views, all values are displayed in one section.

As soon as a user checks a checkbox or changes/enters a value in the input field, this triggers a JavaScript function that sends the input to the server. The input is then stored in a ”redis”-database. This happens without the need for the user to explicitly save his input.

If the user clicks on the link for a regulatory document, a new tab opens on the left side, where the drawing is displayed by default. The tab shows the whole regulatory document. However, if search terms have been provided, these are displayed above the document. The user can click on each of these search terms. The document skips than to the respective page. Additionally the user can always click on ”Start of ISO document”, which brings him back to page one of the document.

If the user wants to upload a new drawing, he/she can simply click on the ”ACDP” logo in the left corner, see figure 4.16 to return to the starting page.

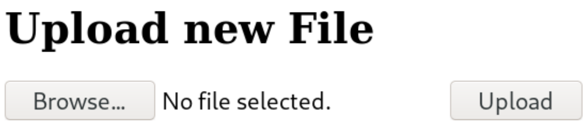
Figure 4.16: Left Corner of the User Interface Showing the CDP Logo and the Tab Containing the Drawing.

4.3.2 Technical Implementation

As mentioned above the web application was implemented using Flask. The first step is to provide a starting page for the user. This is done by providing a simple HTML file that allows the user to upload a new drawing and thereby start the whole extraction process. The drawing is uploaded and stored on the server. The drawing can only be uploaded as PDF. Otherwise the user will be redirect to the starting page. As soon as the drawing is uploaded, a ”uuid”, unique user identification, is generated as well. This ”uuid” is a random number and identifies the drawing throughout the extraction process and also serves as identifier in the ”redis” database. After this is done, the extraction implementation is called, by making a subprocess call which passes all parameters to the extraction program. As next step the libary PyPDF2 [57] is used to get the width and height, as well as the orientation, of the drawing. This will be used later for the highlighting of the values and views. Additionally the PDF file is converted to an image using the function "pdftoppm", which is part of the "poppler-utils" as well. This conversion is done as it is easier to display an image than a PDF in the browser. The extraction process stores the result in "redis", using the "uuid" as the identifier. Therefore the next step for the web application is to get the stored elements from "redis". The extracted elements regarding the general tolerance, the dimensions and the views are stored separately. All of these values are retrieved and stored in variables. Additionally the "config.json" is loaded, which contains all of the search terms for the regulatory documents. Subsequently the dimensions are looped. Firstly, it is checked if they include symbols mentioned in the configuration file. If they do, the search terms including the page numbers are then converted into a dictionary and base64-encoding, to bypass encoding difficulties using JavaScript and HTML. Afterwards, still in the loop, the dimensions are checked for the nominal value, and how many numbers and decimal places it includes. This is necessary to adjust the placeholder for the input field. The coordinates of the dimensions are converted from an array to a string, to bypass encoding issues. Afterwards HTML code for this element in the loop is generated, including dimensions, search terms and coordinates. Lastly, HTML code is added that contains the link to the general regulatory documents. After the loop is finished, all of the data i.e. name of the image,the HTML code, the links to the ISO documents and more information is passed as variables to an HTML template. This HTML file manages the filling in of the values into the respective variables as well as the user interaction. If a user enters a value, a JavaScript function is triggered that sends an AJAX-message to the server-side, where another function handles the interaction with "redis". Additionally, if the user clicks on a value, the view as well as the value itself are highlighted on the drawing. This is managed by two JavaScript functions that are triggered by a click and add/remove an additional CSS-layer, that is put on top of the view and value coordinates, that are stored in the HTML code as additional elements ("data-coords"). The layer that highlights the view area has an opacity of "0.2", whereas the rectangle enclosing the values itself has an opacity of "0.4". To display the highlights on the image correctly, some adaptions have to be made. We already retrieved the width, the height and the orientation of the PDF. That is necessary to adapt the coordinates, initially extracted from the PDF-file, to fit the image displayed here. Each time the user clicks on a value, the current coordinates and size of the image are queried, as this can change if the window size of the browser is adapted. To get the "x" and "y" value of the starting point of the rectangle, two different functions are required. The first one is used when the orientation is "landscape", that means that the PDF is rotated. This function takes the original coordinates multiplied by the height of the image container, divided by the width of the original PDF to calculate the new x-value. Same goes for the new y-value, where the original y-value is multiplied by the width of the drawing container, divided by the original height. If the PDF was not rotated, the calculation are easier and just consist of multiplying the original x-value by the width of the drawing container and divide it by the original width, as well as multiplying the original y-value by the height of the drawing container and multiply it with the original height. Additionally to the x and y-coordinates we need to calculate the relative height and width of the enclosing rectangle. This is similarly done by dividing the original width by the original height and multiplying it with the current width of the container. This is multiplied by the factor 1.4 to enlarge the width of the highlighting rectangle, just for visual purposes. The same is done for the height, where the original height is divided by the original width and multiplied by the current height and the factor 1.4.

4.4 Demonstration

The process begins at the starting page, see figure 4.17, where the user can select the drawing and uploads it to the server.

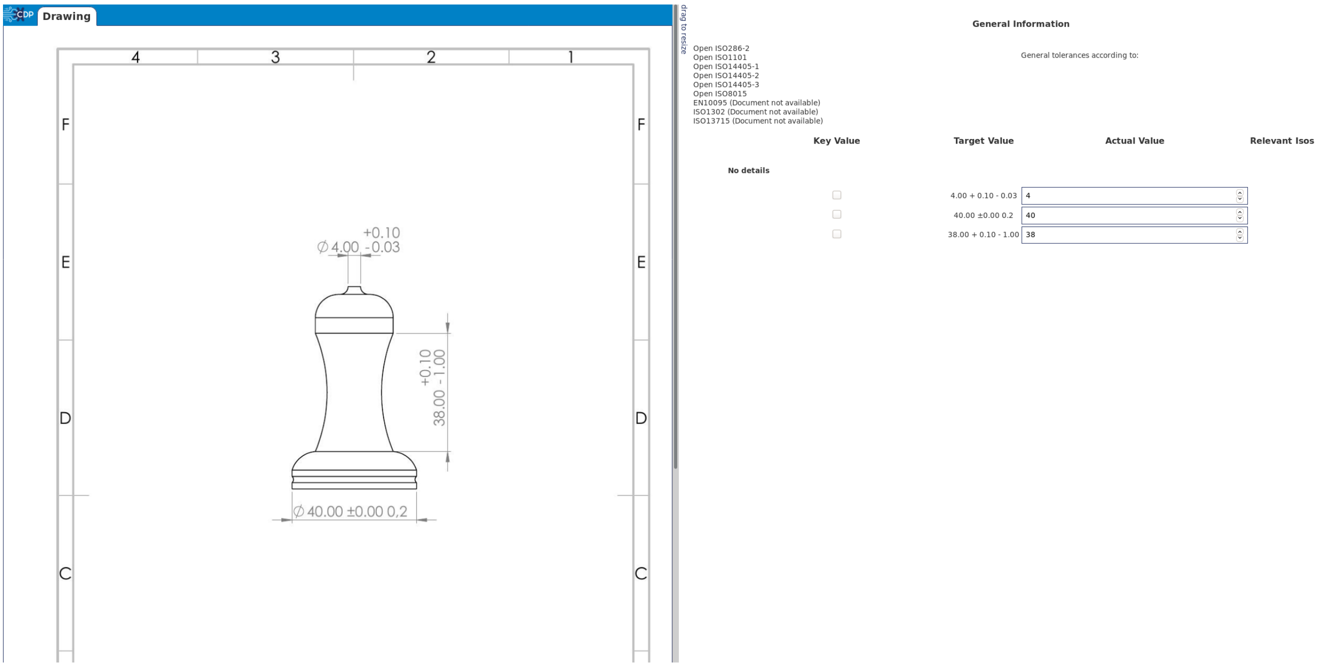


Upload new File

No file selected.

Figure 4.17: Starting Page.

The extraction process then starts and shows the user the extracted dimensions, the additional information and the drawing itself. Figure 4.18 displays the result.



General Information

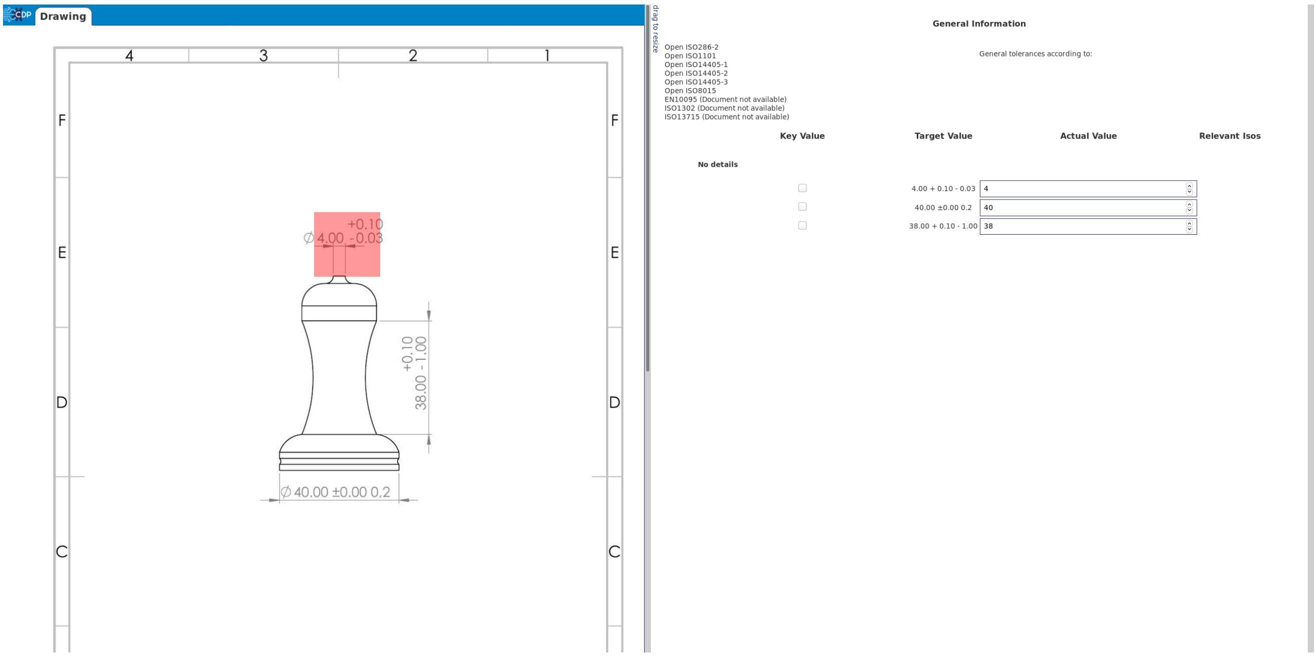
Open ISO286-2
Open ISO1101
Open ISO14405-1
Open ISO14405-2
Open ISO14405-3
Open ISO8015
EN10095 (Document not available)
ISO1302 (Document not available)
ISO13715 (Document not available)

General tolerances according to:

	Key Value	Target Value	Actual Value	Relevant Isos
No details				
	□	4.00 + 0.10 - 0.03	4	
	□	40.00 ±0.00 0.2	40	
	□	38.00 + 0.10 - 1.00	38	

FIGURE 4.18: Overview of User Interface in More Detail.

The user can now click on each row depicting the values, the respective value will be highlighted in the drawing as well as the view it belongs to, if the drawing is partitioned in different views. This can be seen in figure 4.19



General Information

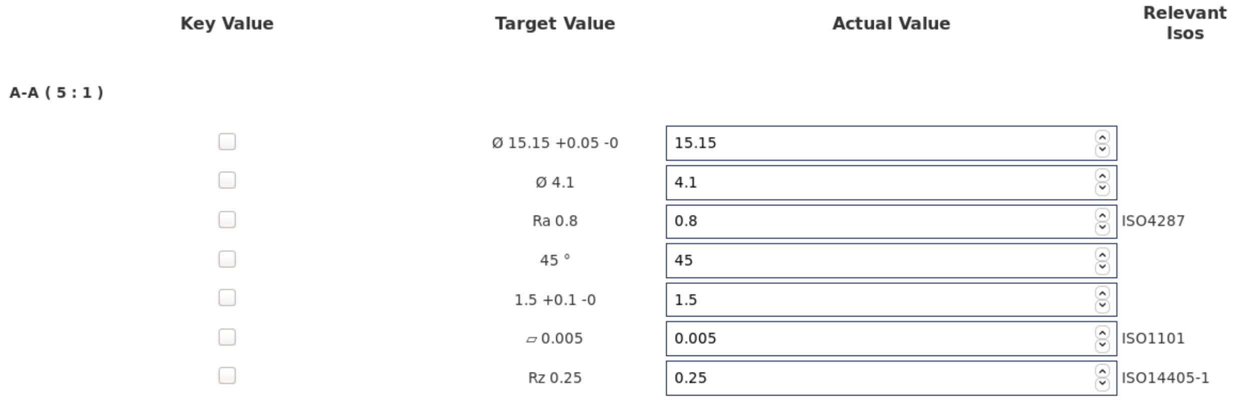
Open ISO286-2
Open ISO1101
Open ISO14405-1
Open ISO14405-2
Open ISO14405-3
Open ISO8015
EN10095 (Document not available)
ISO1302 (Document not available)
ISO13715 (Document not available)

General tolerances according to:

	Key Value	Target Value	Actual Value	Relevant Isos
No details				
	□	4.00 + 0.10 - 0.03	4	
	□	40.00 ±0.00 0.2	40	
	□	38.00 + 0.10 - 1.00	38	

FIGURE 4.19: Highlighted Value and View.

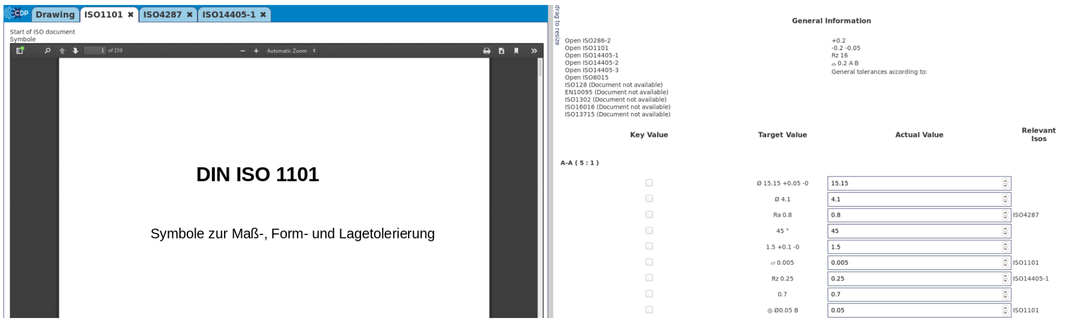
The nominal value is already set as a placeholder in the input field, so the user can change the values up and down or input an entirely different value if necessary. Next to the input field are the relevant regulatory documents linked. This is not the case in the image shown above, but can be seen in figure 4.20, where only part is shown, as the rest of the image is subject to copyright restrictions. This figure also shows how different sections are displayed, here we can see section ”A-A”, as well as the respective scale ”5:1”. The user can click on the links if he or she has to look up a regulation.



Key Value	Target Value	Actual Value	Relevant Isos
A-A (5 : 1)			
□	Ø 15.15 +0.05 -0	15.15	
□	Ø 4.1	4.1	
□	Ra 0.8	0.8	ISO4287
□	45 °	45	
□	1.5 +0.1 -0	1.5	
□	▱ 0.005	0.005	ISO1101
□	Rz 0.25	0.25	ISO14405-1

Figure 4.20: Extract of a Drawing, Showing Values and Links.

In figure 4.21 we can see, how the linked documents are displayed. As the user clicks on a link, a new tab on top of the drawing is opened, showing the document and the search terms.



Start of ISO document
Symbole

DIN ISO 1101

Symbole zur Maß-, Form- und Lagetolerierung

General Information

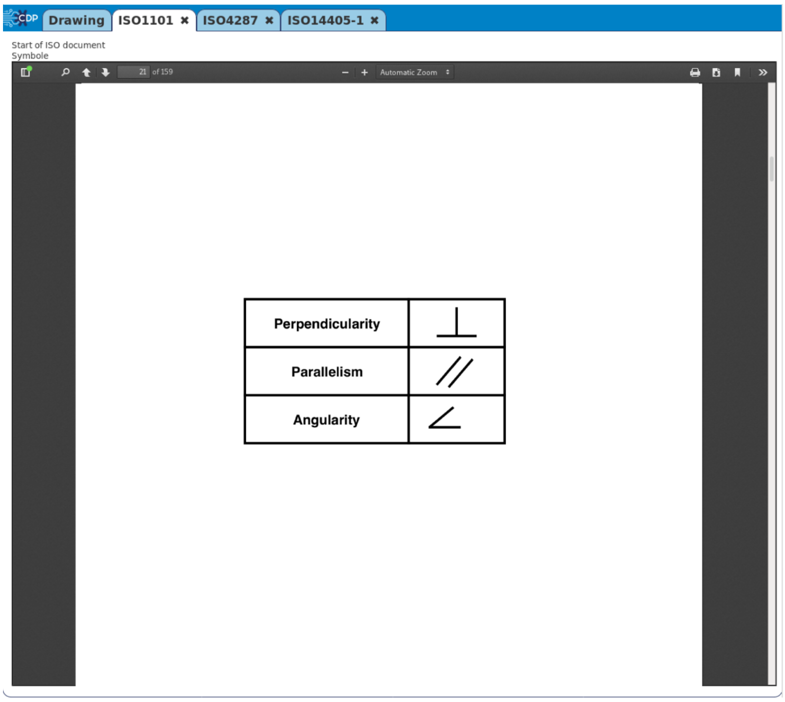
Open ISO286-2
Open ISO1101
Open ISO14405-1
Open ISO14405-2
Open ISO14405-3
Open ISO8015
ISO128 (Document not available)
EN10095 (Document not available)
ISO1302 (Document not available)
ISO16016 (Document not available)
ISO13715 (Document not available)

+0.2
-0.2 -0.05
Rz 16
⌓ 0.2 A B
General tolerances according to:

Key Value	Target Value	Actual Value	Relevant Isos
A-A (5 : 1)			
□	Ø 15.15 +0.05 -0	15.15	
□	Ø 4.1	4.1	
□	Ra 0.8	0.8	ISO4287
□	45 °	45	
□	1.5 +0.1 -0	1.5	
□	▱ 0.005	0.005	ISO1101
□	Rz 0.25	0.25	ISO14405-1
□	0.7	0.7	
□	◎ Ø0.05 B	0.05	ISO1101

Figure 4.21: Showing the Linked Regulatory Documents and the General Tolerance.

If we click on the search term ”Symbole”, the document jumps to the specific section, see figure 4.22. To go back to the start of the document a click on ”Start of ISO document” is enough.

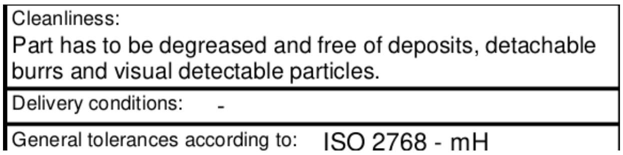


Start of ISO document
Symbole

Perpendicularity	⊥
Parallelism	//
Angularity	∠

FIGURE 4.22: Navigating Through the Regulatory Document.

Figure 4.24 shows the general information for another drawing, which can not be shown in full for copyright issues as well. However, in this drawing is a specific tolerance class noted as can be seen in figure 4.23.

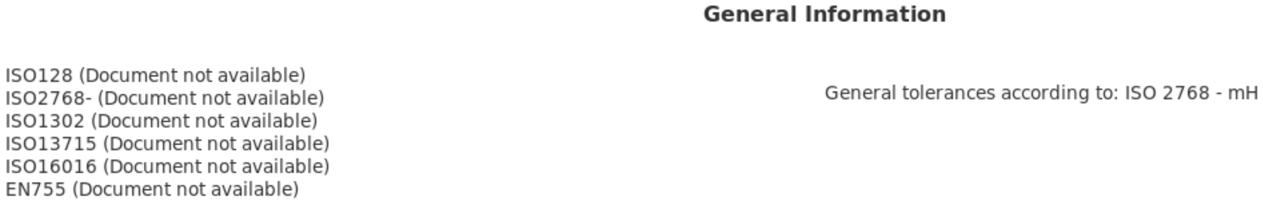


Cleanliness:
Part has to be degreased and free of deposits, detachable burrs and visual detectable particles.

Delivery conditions: -

General tolerances according to: ISO 2768 - mH

FIGURE 4.23: General Tolerance Class Noted in the Drawing.



General Information

ISO128 (Document not available)
ISO2768- (Document not available)
ISO1302 (Document not available)
ISO13715 (Document not available)
ISO16016 (Document not available)
EN755 (Document not available)

General tolerances according to: ISO 2768 - mH

FIGURE 4.24: Tolerance Class and Regulatory Documents.

4.5 Combined Workflow

The figure 4.25 depicts the rough workflow of the prototype. The figure is partitioned into three columns, the user, the web application (which includes server and client side) and the extraction process itself. The process is displayed only roughly, the main goal being the interaction visualization of the different parts. As can be clearly seen, the interaction between the web application and the extraction program limits itself to the initial call to start the extraction process. Any further interaction is done by using the "redis" database as an intermediate step.

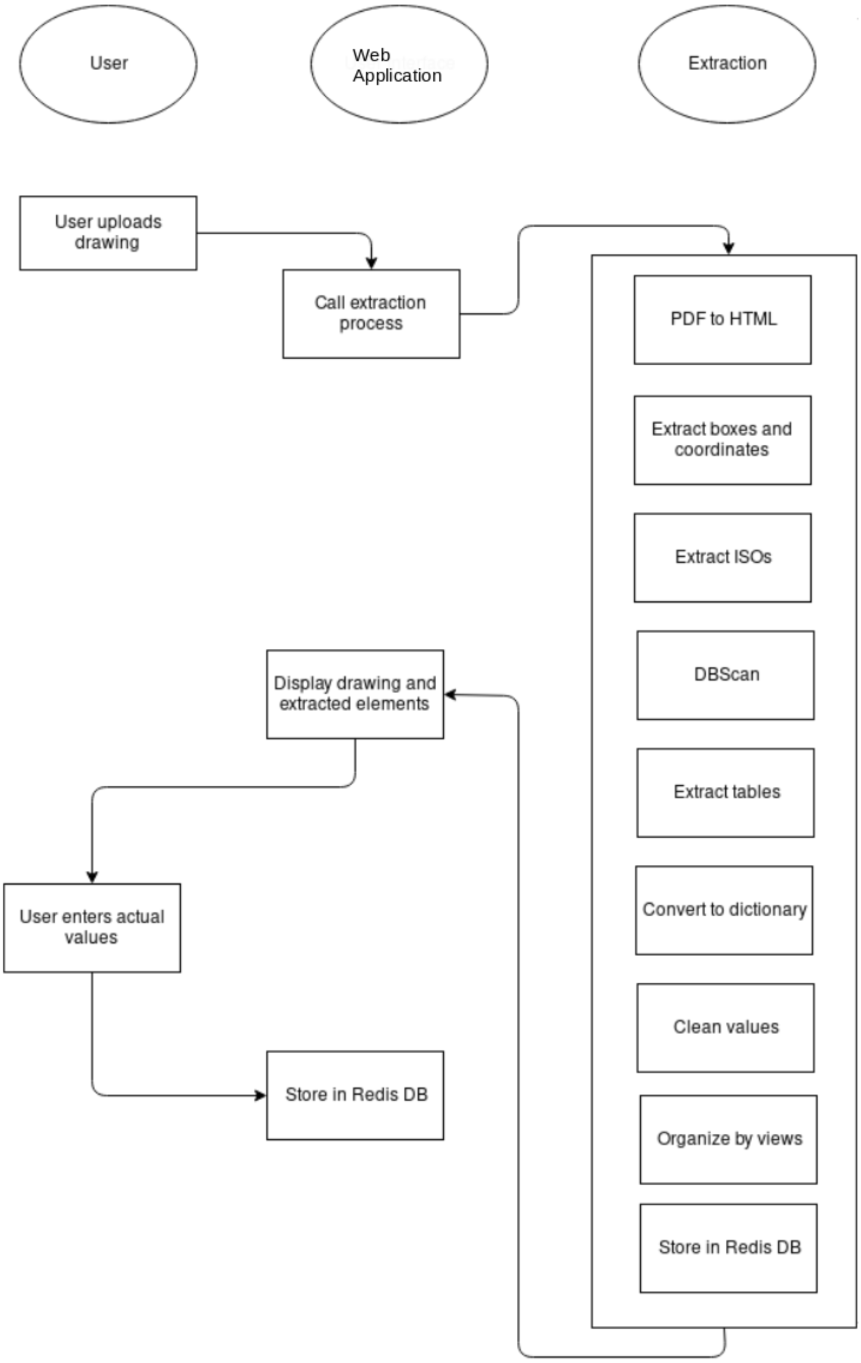


FIGURE 4.25: Overview of Internal Workflow.

This rough overview does not take into account the process flow inside the web application, including the communication from server to client and vice versa.

The here described prototype enables the user to have a clear overview of all dimensioning requirements, as well as additional regulations, and provides input fields to log actual measurements. However, while implementing this prototype other questions arose that can be part of future work. Firstly, the values in this prototype are sorted by the views they belong to. In the future a more refined sorting-approach could be implemented that includes the importance of the values. This is quite a complex issue, as the importance of dimensions differs for each workpiece depending on the functionality of the workpiece. Nevertheless, it could be possible to provide an ontology or any other kind of domain description which provides a ranking for the most important dimensions depending on which kind of workpiece is depicted by the drawing. Additionally, in future work the user could have more interaction possibilities to adapt the user interface according to his/her need. ´

Chapter 5

Evaluation

After the prototype was developed, two different kinds of evaluation, qualitative and quantitative, have been conducted. The primary form of evaluation consisted of a quantitative analysis. In a second step, an expert interview, to check if the implementation should be adapted to fit the need of real world applications, was conducted. Therefore the quantitative analysis focuses mainly on the accuracy of the extraction process, whereas the interview addresses mainly the user interface and the relevance of the implementation in the industry.

5.1 Quantitative Analysis

Before an in-depth-analysis could be conducted, measurements had to be specified. As the main key measure a confusion matrix was used. A confusion matrix is mainly used in the field of classification problems for evaluating the performance of a classifier. [75] defines it as a matrix consisting of two dimensions, the first dimension being the original class of an element and in the other dimension being the class assigned by the classification algorithm. Even tough, it is mainly used for classification, it can be adapted to fit the needs for this thesis.

Figure 5.1 shows the adapted version. The matrix consists of four fields and is divided into the dimensions ”extracted” and ”relevant”, which can either be ”yes” or ”no”. ”Relevant” in this context means that it is a dimension (or a dimension set), not other textual information. The keyword ”extracted” refers to if it was recognized in the extraction process. The upper left field which consists of ”yes” in both dimensions, can also be seen as the ”true positive” field, as all values here have been extracted and are relevant. The upper right field is the equivalent of ”false negative” as these values are relevant, but have not been extracted. The lower left field consists of values that have been extracted but are not relevant, which can also be described as "false positive". The lower right field comprises values that are not relevant and have also not been extracted, i.e. "true negative" values.

Other, related, measurements, are precision and recall, which are commonly used to evaluate information extraction systems. [76] defines precision as the number of relevant documents that have been retrieved divided by the number of retrieved documents. This can also be expressed as the number of true positive elements divided by the number of all extracted elements. The recall measure is defined as being the the number of relevant documents that have been retrieved divided by the number of all relevant documents [76], or in terms of the confusion matrix as the number of true positive elements divided by the number of all relevant elements. The drawings were checked manually for the number of relevant elements.

When counting the correctly extracted elements, only elements where the whole dimension set was extracted was seen as correct, However if more than one dimension set was in one row, the dimension sets were still counted as correct. For the extracted elements the elements seen in the user interface are counted, so these elements have been extracted and also post-processed. So any errors could have occurred in the pre-processing, extraction or post-processing phase. As views, only views with a title are included, as without a title, it would need more sophisticated object detection algorithms, including recognition of graphical elements, to detect these.

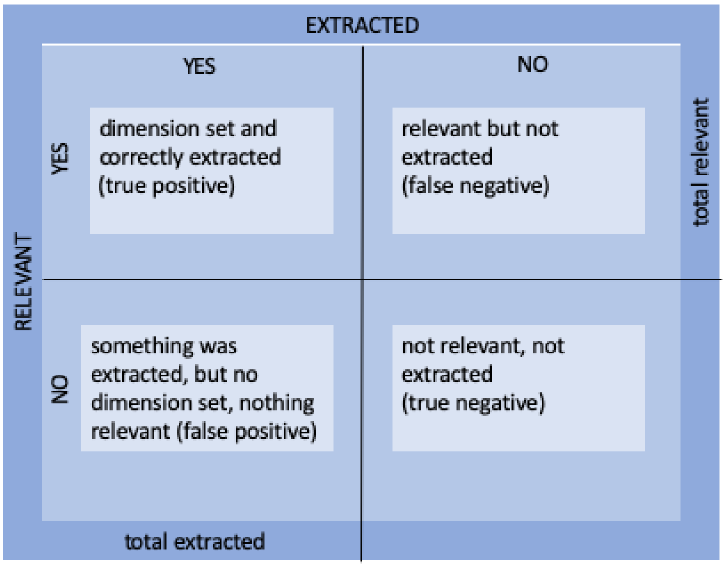

EXTRACTED
YES
NO
RELEVANT
YES
NO
dimension set and correctly extracted (true positive)
relevant but not extracted (false negative)
something was extracted, but no dimension set, nothing relevant (false positive)
not relevant, not extracted (true negative)
total relevant
total extracted

FIGURE 5.1: Confusion Matrix Used for Quantitative Evaluation.

In addition to these indicators other describing factors are documented as well. These factors are the number of views, the amount of blocks and lines in the corresponding HTML file as well as the number of clusters.

Therefore the following indicators are gathered from each drawing:

- confusion matrix
- precision value
- recall value
- total amount of blocks and lines (extracted from HTML)
- number of views
- number of clusters

For the prototype six drawings were examined in regard to these indicators. The evaluation results can be seen in the next sub sections. Additionally, to the descriptive elements and the confusion matrix, the user interface for the respective drawing will be depicted. However, as some of the drawings are copyright protected, the values are blurred. The drawings are ordered by complexity (defined by number of blocks and words), starting with the least complex drawing. The two drawings that are not copyright protected, are available at the GitHub repository (see Appendix A).

5.1.1 Results Drawing 1

The first drawing was already used for the demonstration. It is a simple drawing with few values as can be seen in table 5.1 and correspondingly all values are detected in the extraction process.

Views	Blocks	Words	Cluster
1	37	100	36

TABLE 5.1: Statistics of Drawing ”Laeufer”.

- Precision value: 1
- Recall value: 1

These values illustrate what can already be seen by looking at the matrix (figure 5.2), that all values that have been extracted are relevant and that all relevant values are recognized.

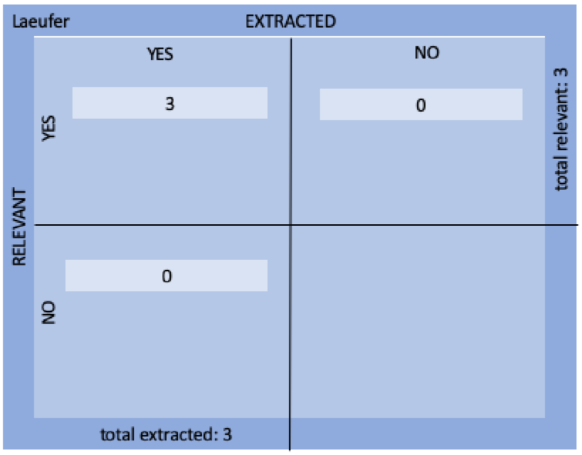
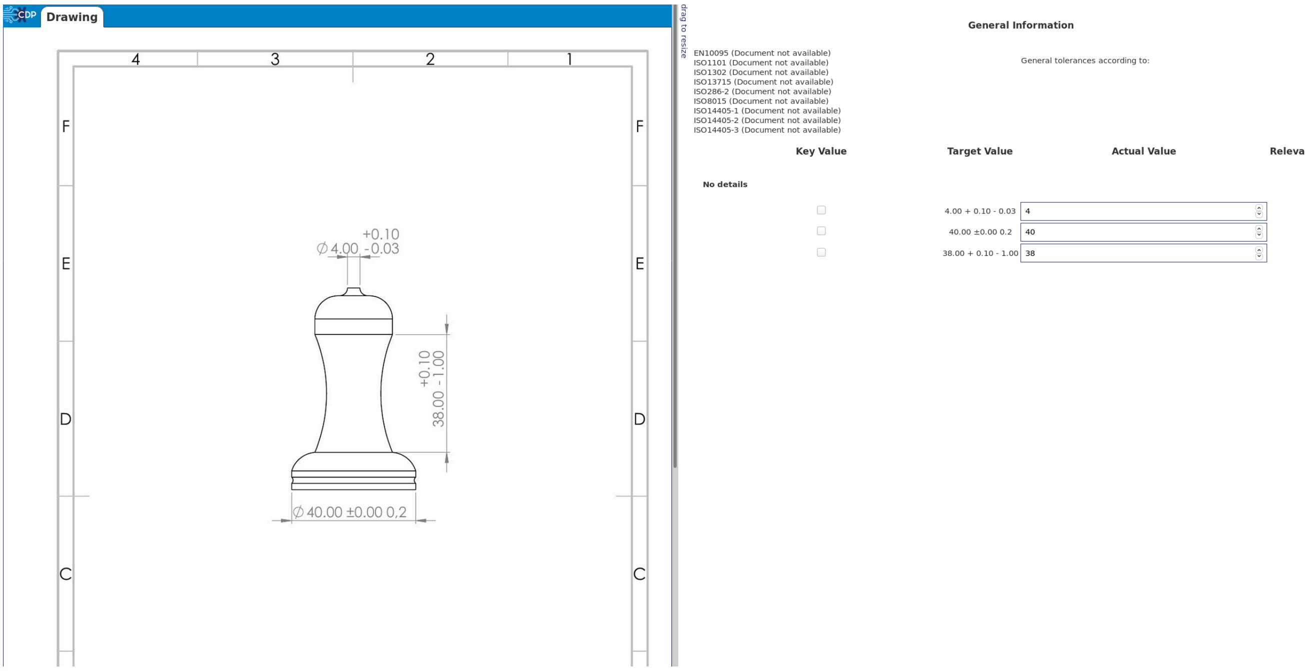


FIGURE 5.2: Confusion Matrix for ”Laeufer”.



General Information

EN10095 (Document not available)
ISO1101 (Document not available)
ISO1302 (Document not available)
ISO13715 (Document not available)
ISO286-2 (Document not available)
ISO8015 (Document not available)
ISO14405-1 (Document not available)
ISO14405-2 (Document not available)
ISO14405-3 (Document not available)

General tolerances according to:

No details

Key Value	Target Value	Actual Value	Releva
□	4.00 + 0.10 - 0.03	4	
□	40.00 ±0.00 0.2	40	
□	38.00 + 0.10 - 1.00	38	

Figure 5.3: User Interface for "Laeufer".

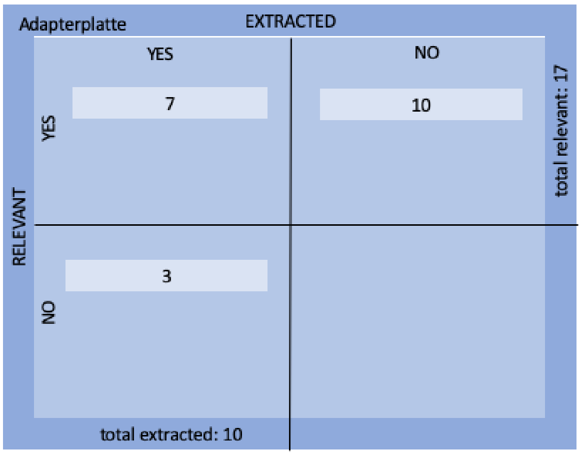
The user interface (figure 5.3) as already depicted in section 4. All values are depicted correctly. As there is only one view, no sectioning is done.

5.1.2 Results Drawing 2

The second drawing is a similarly uncomplex drawing, see table 5.2. However, the extraction process was less successful than for drawing 1, see the confusion matrix (figure 5.4).

Views	Blocks	Words	Cluster
1	72	104	66

Table 5.2: Statistics of Drawing "Adapterplatte".

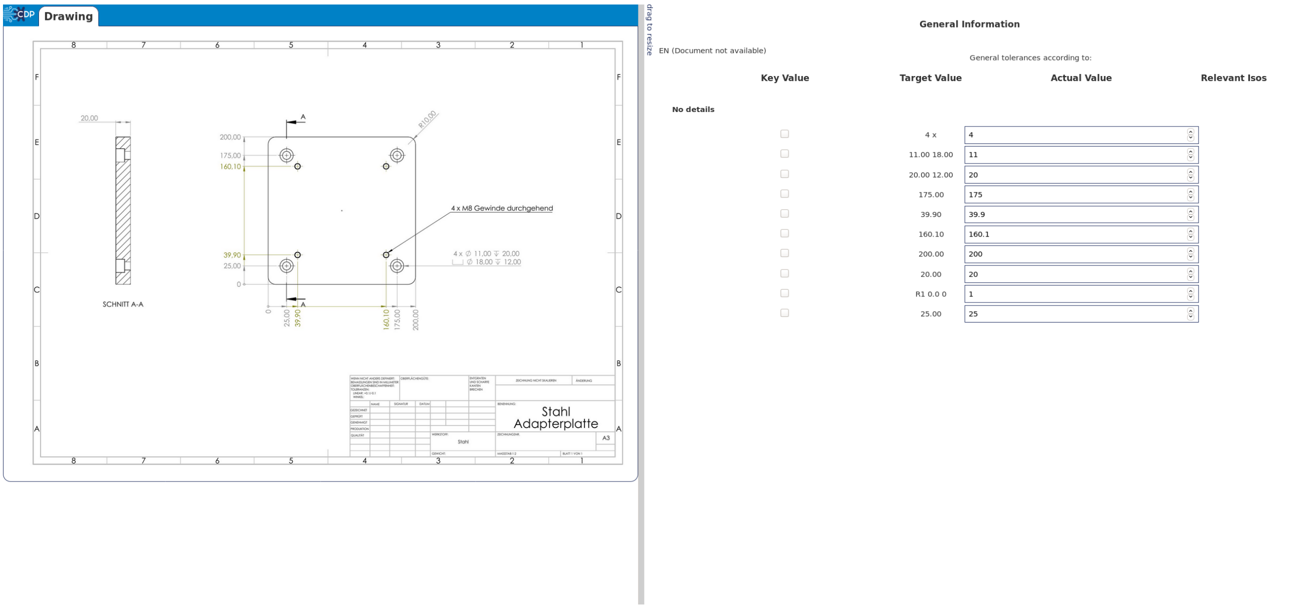


Adapterplatte	EXTRACTED: YES	EXTRACTED: NO	
RELEVANT: YES	7	10	total relevant: 17
RELEVANT: NO	3		
	total extracted: 10		

FIGURE 5.4: Confusion Matrix for "Adapterplatte".

- Precision value: 0.7
- Recall value: 0.41

These values mean that 70% of the extracted values are relevant and only over 40% of the total relevant elements have been extracted at all. This is a surprisingly inaccurate result for being such a simple drawing. One reason why it might be this low, is that it is a slightly different format, as there are no dimension sets, only single nominal dimensions are given. In addition, it could be possible that the EPS value has to be adapted, as maybe more clusters would have led to better results.



General Information

EN (Document not available)

General tolerances according to:

No details

Key Value	Target Value	Actual Value	Relevant Isos
□	4 x	4	
□	11.00 18.00	11	
□	20.00 12.00	20	
□	175.00	175	
□	39.90	39.9	
□	160.10	160.1	
□	200.00	200	
□	20.00	20	
□	R1 0.0 0	1	
□	25.00	25	

FIGURE 5.5: User Interface for "Adapterplatte".

The user interface as seen in figure 5.5 is quite simple for this drawing as it consists of only two views, however as there is only one view title it can still be seen as having only one main view. Accordingly, the drawing is not sectioned.

5.1.3 Results Drawing 3

The third drawing is already more complex, see table 5.3, having multiple views and a wide range of elements. This drawing (see figure 5.7) is copyrighted and therefore all values are blurred in this thesis. The remainder of the evaluation drawings will be blurred as well. The result can be seen in figure 5.6.

Views	Blocks	Words	Cluster
4	126	401	125

TABLE 5.3: Statistics of Drawing "GV12".

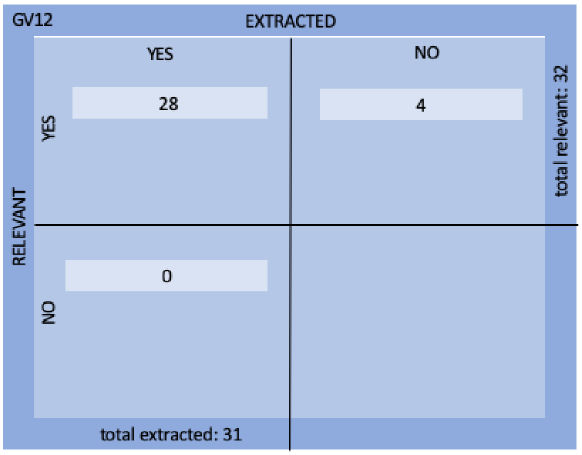


FIGURE 5.6: Confusion Matrix for "GV12".

- Precision value: 0.9
- Recall value: 0.875

Even tough this drawing is more complex, the accuracy is higher. The precision value of 0.90 means that 90% of all extracted values are relevant and the recall value says that almost 88% of all relevant values have been recognized correctly.

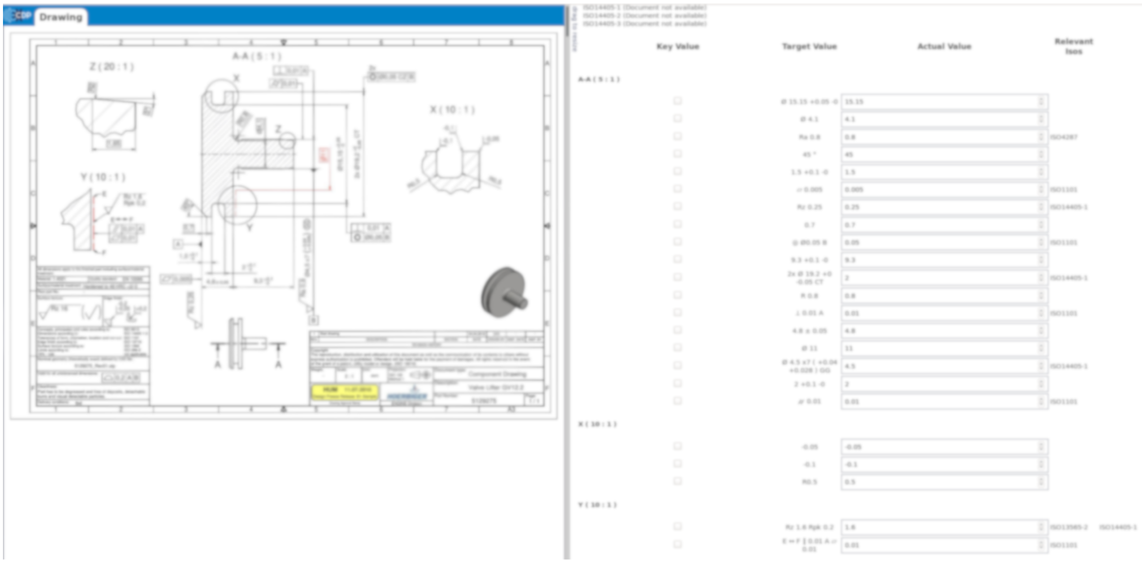


Figure 5.7: User Interface for ”GV12”.

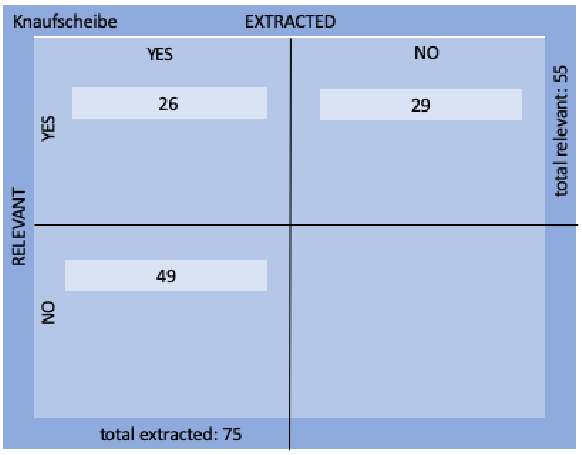
The generation of the user interface, see figure 5.7, worked nicely, all views as well as the additional regulatory documents have been extracted properly.

5.1.4 Results Drawing 4

The fourth drawing is also from a company and therefore blurred. However, as it is from another company as the one before, it also has a different structure and layout. It is also a more complex drawing with even more elements, as can be seen in table 5.4. The respective results can be seen in figure 5.8.

Views	Blocks	Words	Cluster
4	212	296	192

Table 5.4: Statistics of Drawing ”Knaufscheibe”.



Knaufscheibe		EXTRACTED		
		YES	NO	
RELEVANT	YES	26	29	total relevant: 55
	NO	49		
		total extracted: 75		

FIGURE 5.8: Confusion Matrix for "Knaufscheibe".

- Precision value: 0.53
- Recall value: 0.89

The extraction process yielded a precision value of 0.53, which means that 53% of all extracted values are actual dimension elements, however almost 90% of all relevant values have been extracted. This means that a lot of unnecessary elements have been recognized as being a dimension (set). This is probably due to the fact that it is another layout than what was used for developing the prototype, so more adapting has to be done in future work to include other structure and layouts as well.

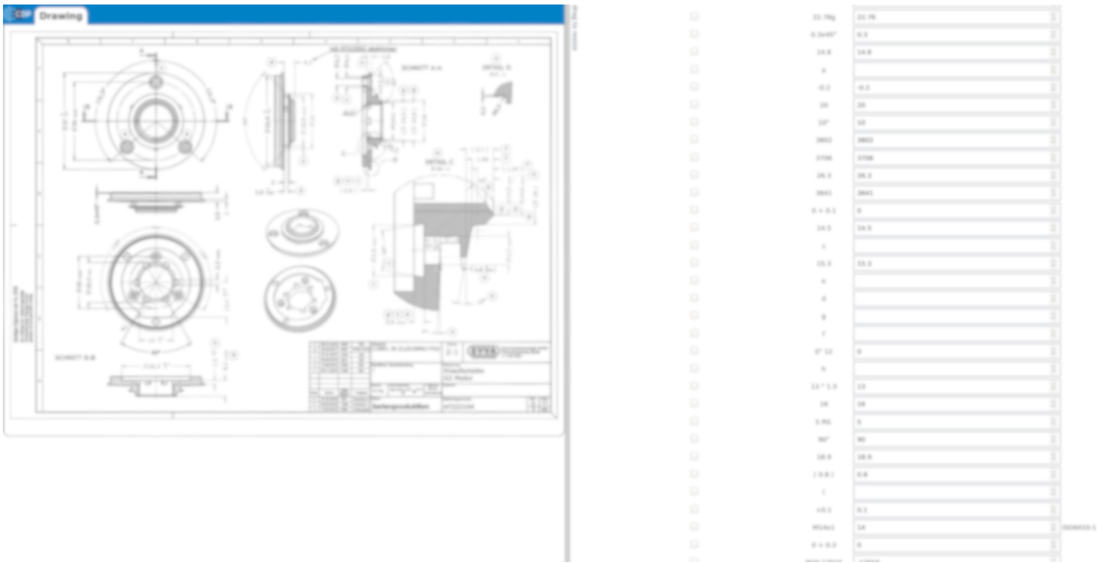


FIGURE 5.9: User Interface for "Knaufscheibe".

The user interface consists of only one view, as can be seen in figure 5.9, as the view titles are in another format as well (which can be easily adapted in future work). Otherwise, all values can be highlighted by clicking on them, so the highlighting works for different layouts.

5.1.5 Results Drawing 5

This drawing has a similar structure to drawing 4, but is a little bit more complex in terms of numbers of blocks and words, see table 5.5. The extraction results can be seen in figure 5.10.

Views	Blocks	Words	Cluster
4	218	337	204

Table 5.5: Statistics of Drawing "Knaufachse".

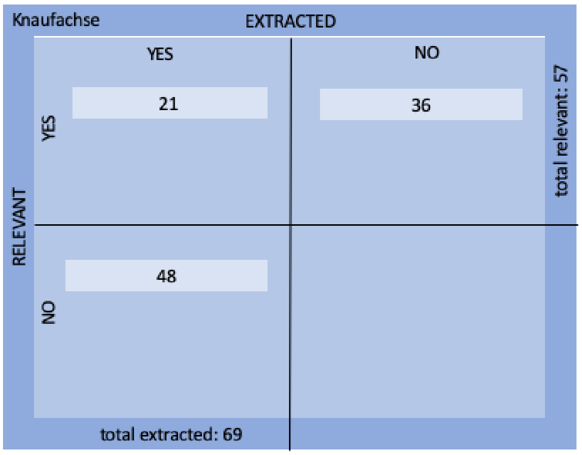
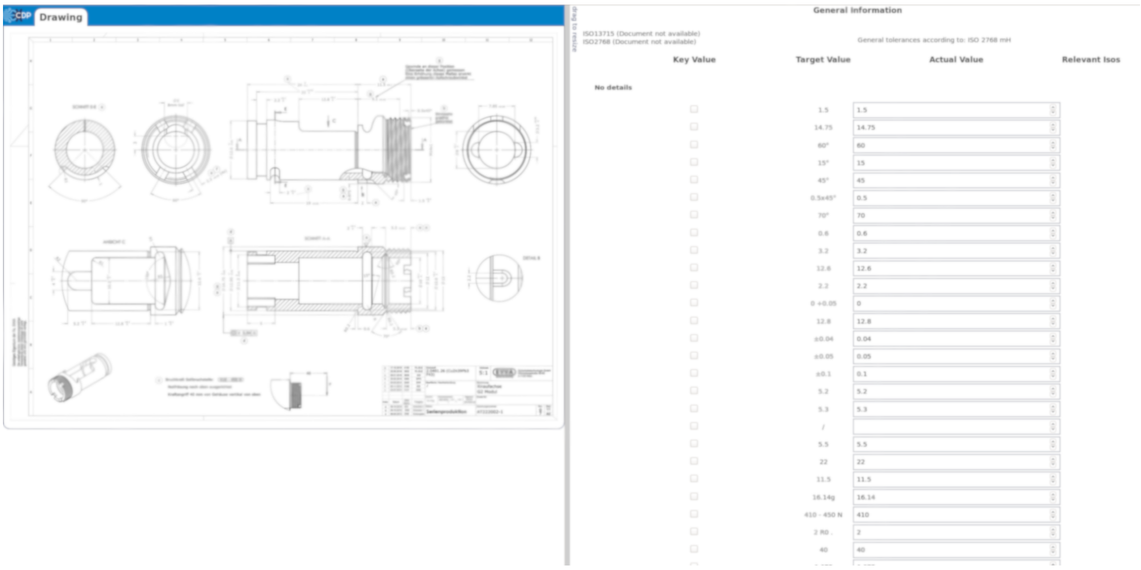


Figure 5.10: Confusion Matrix for "Knaufachse".

- Precision value: 0.3
- Recall value: 0.58

The precision value is the lowest so far, as only 30% of all extracted elements are relevant, the recall value is a little higher with 0.58, meaning that 58% of all relevant values have been recognized. As in the drawing before, these low values are probably due to lacking adaption to this specific structure and could be improved in future work.



General Information

ISO13715 (Document not available)
ISO2768 (Document not available)

General tolerances according to: ISO 2768 mH

Key Value	Target Value	Actual Value	Relevant Isos
No details			
□	1.5	1.5	
□	14.75	14.75	
□	60°	60	
□	15°	15	
□	45°	45	
□	0.5x45°	0.5	
□	70°	70	
□	0.6	0.6	
□	3.2	3.2	
□	12.6	12.6	
□	2.2	2.2	
□	0 +0.05	0	
□	12.8	12.8	
□	±0.04	0.04	
□	±0.05	0.05	
□	±0.1	0.1	
□	5.2	5.2	
□	5.3	5.3	
□	/		
□	5.5	5.5	
□	22	22	
□	11.5	11.5	
□	16.14g	16.14	
□	410 - 450 N	410	
□	2 R0 .	2	
□	40	40	

Figure 5.11: User Interface for "Knaufachse".

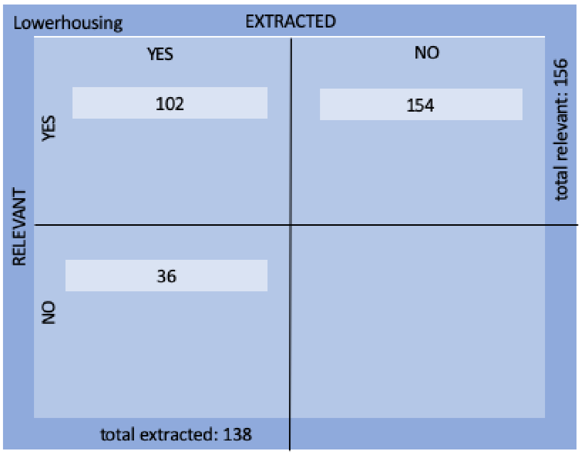
As in drawing 4, the views were not recognized, however the highlighting works accurately. The user interface is depicted in figure 5.11.

5.1.6 Results Drawing 6

The last drawing that was evaluated is also the most complex one, as it includes 14 views and over 800 words, as can be seen in table 5.6. The structure is challenging for the prototype as some elements are overlapping or stacked upon each other. The corresponding results can be seen in figure 5.12.

Views	Blocks	Words	Cluster
14	330	827	250

Table 5.6: Statistics of Drawing "Lowerhousing".



Lowerhousing		EXTRACTED	
		YES	NO
RELEVANT	YES	102	154
	NO	36	

total relevant: 156

total extracted: 138

FIGURE 5.12: Confusion Matrix for ”Lowerhousing”.

- Precision value: 0.74
- Recall value: 0.65

Even tough the drawing is highly complex, the precision value is over 0.7, which means that more than 70% of all extracted values are dimension related, and also 65% of relevant elements have been recognized correctly. Some of the values were not recognized because they are stacked on top of each other, so the algorithm cannot not assign them properly to a dimension set. This problem however should be solved, if all drawings are constructed in a way that strictly adheres to the ISO standards, as overlapping elements are not allowed according to ISO regulations.

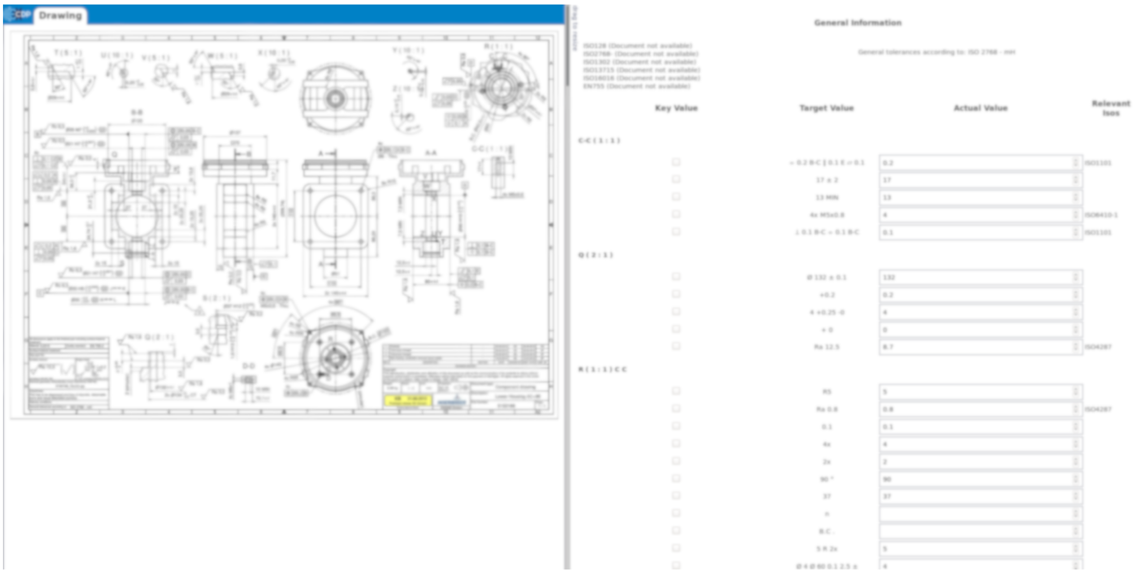


FIGURE 5.13: User Interface for ”Lowerhousing”.

The user interface is depicted in figure 5.13.The view titles were correctly extracted. However, the highlighting does not work for all views. Only six views (out of the 14) have correct borders, as for the others it was too difficult for the prototype to distinguish were a new view starts. This may be due to the complexity of the drawing.

5.1.7 Result Summary

An overview of the evaluation results can be seen in the table 5.7.

Drawing	# Total Relevant	# Total Extracted	# Cluster	Precision	Recall
1	3	3	36	1	1
2	17	10	66	0.7	0.41
3	32	31	125	0.9	0.86
4	55	75	192	0.53	0.89
5	57	69	204	0.3	0.58
6	156	138	250	0.74	0.65

Table 5.7: Overview of Drawings Used for the Quantitative Evaluation.

- Average precision: 0.70
- Average recall: 0.73

The average precision is 0.70, which means that overall 70% of the extracted values, are dimension (sets) and therefore relevant. The average recall is 0.73, therefore overall 73% of relevant elements have been extracted. The extracted but not relevant elements are mostly punctuation marks, parts of dimension sets that have not been extracted correctly or other parts of the textual elements that are not relevant for the dimensioning. The fluctuation in the values is quite high, as the precision value differs between 30% and 100% and the recall value varies between 41% and 100%. These fluctuation can be due to the different complexities of the drawings, as well as varying structures and layouts. This can be seen when looking at the results for the drawings 4 and 5, which have a quite different structures to the rest of the drawings, and the precision values are lowest. The number of clusters is mostly higher than the number of extracted elements, which shows that after clustering, the post-processing was able to remove additional information. Additionally some elements, that are originally separate are extracted as one cluster, nevertheless if they are extracted correctly they are counted towards the accurately extracted elements, as dividing them can be easily done in post-processing. Additionally, to the varying layouts, the EPS value could also have an impact on the accuracy of the results. Currently only two different EPS values are used, which are decided by the number of words in a drawing. This process can, in future work, be adapted, to see which factors influence the EPS value in an optimal way.

5.2 Expert Interview

Additionally, to the quantitative analysis, a domain expert was consulted, regarding the usability and relevance of this prototype in the industry. [1] Dr. Florian Pauker, Digitalization Operations Manager at EVVA Sicherheitstechnik GmbH, was chosen as the interview partner, as he has significant experience in the production domain, specifically in regard to digitalization of production processes.

Two main aspects were discussed:

- In which ways is extraction of dimensioning requirements useful and relevant in the industry?
- How can the prototype be improved?

Regarding the first question Dr. Pauker mentioned the following points:

- The use case that was applied as base for this thesis, namely using the system as part of simplifying the measuring process for employees, is relevant in the industry.
- Additionally, a foremen could do the first measuring, entering which elements are essential and which are not important. This information is then stored in the database and for the following iterations only the important elements are shown to the employees, so they can focus on the essential parts.
- The automatic generation of measuring programs is also a current topic in the industry. The prototype could extract quality relevant information and therefore save the effort of manually preparing the basic information for the measuring program.

The following aspects were mentioned by the expert for adapting the user interface in future work:

- Numbering the extracted dimensions in the drawing as well as on the right side of the user interface for easier matching of the elements
- Different highlights for important elements (e.g. red highlight for essential dimensions, which are defined by the foreman, blue highlights for non essential elements)
- Supporting DXF files, as DXF is a common standard in the industry

[1] See interview protocol at GitHub `https://github.com/bscheibel/masterthesis_webapp/blob/master/protocol_interview_florian_pauker`

- Enlarging the supported structures of drawings for the extraction process, also supporting layouts that do not adhere to the norms
- Making the system adaptable by the user
- Developing a measure for quality and compliance with standards for drawings

5.2.1 Deployment Scenarios

Throughout the qualitative analysis, the expert mentioned an additional use case. The use case that served as a basis for this master thesis focuses on quality control of the workpiece. However, this tool could also be used as quality control for the drawing itself. As mentioned in the quantitative analysis not all drawings are useful for the extraction process. It is important that the different values are neatly organized and specifically no values are overlapping or stacked upon each other. If this is the case, the extraction process cannot work properly, but additionally it is also more complex for a human to understand the drawing. Therefore, an iterative drawing process could consist of constructing the technical drawing, checking the quality by using the system, then making corrections in the design if necessary.

Chapter 6

Conclusion

As shown in this thesis, extracting dimensioning requirements from engineering drawings can be achieved, is relevant in the industrial domain for automating quality control but also for checking the quality of engineering drawings itself. Based on the research questions, identified in section 1, this thesis explored solutions regarding these questions by developing a prototype. In the following, answers to each of these questions are given, using the information that has been obtained during the creation of this thesis.

Question 1: How can textual information, specifically dimensioning requirements, be extracted out of an engineering drawing?

In section 3 and section 4 different approaches are discussed and analyzed. The implemented prototype consists of pre-processing to extract HTML elements out of PDF files. HTML is used as it contains coordinates as well as the textual part, which simplifies the process of regrouping and highlighting later on. After the elements are extracted, the dimension sets have to be recomposed which is done using DBSCAN. Afterwards post-processing has to be done, to differentiate between dimension elements and other textual information. In regard to literature, this approach is new as the goal itself, to extract all information regarding dimensioning, has not been pursued yet. Most related work, rather focus on graphical elements or all textual elements, not specifically on everything related to dimensioning, including the title block and additional information. However, the algorithms used, have been mentioned in the literature. For example [7] and other authors used heuristics to gather relevant information. New to this approach is that, only the pre- and post processing contains domain specific knowledge. The clustering process itself only relies on distance measures and no additional information is necessary.

Sub-question 1a: How does the extracted information relate to regulatory documents? Regulatory documents provide further information and specification for dimensioning and tolerances. For example a tolerance class can be specified in the drawing. This tolerances class then determines the tolerances for all dimensions where no explicit tolerance is stated. For the use case used as basis for the prototype, this information is useful for the employee conducting the quality control, as it may be necessary to check the regulatory standards to clarify quality requirements.

Sub-question 1b: How can relevant information from regulatory documents be identified and extracted? In the process of extracting information from the drawing, all keywords pointing to regulatory documents are extracted. These keywords are mainly located in the title block or additional tables. This information is then used in conjunction with an ontology to provide the user with all necessary information. As most of the regulatory information is a PDF file, the extraction can be quite complex. Different approaches are tried and analyzed in chapter 4. The approach that was then chosen for the prototype consist of constructing an ontology containing information about which references point to which regulatory file, and search terms, which can be used to search the file for relevant information. References in this case comprise labels, explicitly mentioning an ISO standard, as well as symbols which are used for referring to specific features like surfaces, geometrical tolerances and more. In the user interface all regulatory documents, mentioned in the drawing, are linked, allowing the user to readily look up additional information. Furthermore the user can jump around important sections in the document using the search terms specified in the ontology, for easier use.

Question 2: In which ways is this prototype relevant in the production domain? This issue was mainly answered by the expert interview. A domain expert was questioned regarding the usability of the prototype in the industry. The expert reinforced the assumption that this prototype could simplify the quality control process as well as help further automating the production process as a whole. This can be achieved either by supplying an employee with additional information to streamline manual control, by being used as the basis for a measuring program as part of an automatic quality control or by being feedback for the design engineer of the drawing at the designing phase see sub-question 2b.

Sub-question 2a: How well can the extraction be achieved in terms of accuracy? After the prototype was developed, some key indicators were established to measure the accuracy of the extraction results. These indicators are based on a confusion matrix, which gives an overview over relevant and extracted elements. The key measurements used here are "precision" and "recall". The precision value specifies how many of the retrieved elements are relevant, whereas recall specifies how many of all relevant elements have been retrieved. Both of these values varied among the drawings used for evaluation, ranging from 1 (which means 100%) to 0.3. The average precision was 0.7, which means of all extracted elements 70% are indeed dimensioning requirements and relevant, whereas and the average recall was 0.73, meaning that 73% of all relevant information has indeed been extracted. These numbers are indicators that the extraction process does work, but still more optimization can be done in future work, see section 6.1.

Sub-question 2b: Which alternative application scenarios are possible? Additionally, to the use case that was used as basis for this thesis other use cases can be thought of. First of, the implementation can be used to generate an automatic measurement program, by extracting all information that is relevant for quality control. This would be a more efficient approach than manually extracting this information. A measurement program could then be used to fully automate the quality control process. An application scenario that was explicitly mentioned as particularly relevant by the domain expert, is quality control for the drawings itself. As was observed during development of the prototype, not all drawings adhere to norms regarding distance between elements, naming, and so on. If these rules are not followed, it may get more complicated to extract information, for example because elements can overlap. The implementation described in this thesis could be used to immediately check the extractability of a drawing, while the drawing is still in the designing phase. The design engineer can immediately react and adapt the drawing accordingly, facilitating reading of the drawings for humans as well as for this prototype.

To conclude, extracting textual information from engineering drawings is a relevant issue in the industry and has multiple application scenarios. The development of this prototype showed that it is possible to extract dimensioning information from engineering drawings. However, the accuracy can vary depending on the layout and other factors. Some guidelines for constructing technical drawings can be established that improve the ability to be read automatically:

- The technical drawings have to be in digital PDF, no scans.
- The ISO norms regarding the distances between elements have to be followed to avoid overlapping.
- The different views have to be clearly separated as well to be able to differentiate between views.
- Additionally, the views have to be properly labeled to be recognized by the algorithm.
- A table containing all relevant norms is provided.

- The tolerance class is mentioned in the title block or in an additional table.
- The drawing is either in conventional portrait or landscape format.

If all of the guidelines are followed, even complex drawings should be automatically readable. Additionally, these drawings should also be easier understandable by humans.

6.1 Limitations and Future Research

For this prototype some assumptions were made. Firstly, the drawing has to be in digital PDF format. Secondly, there is a wide variety of types of technical drawings, all of them having different layouts. This thesis here focused on part drawings. Therefore, the prototype might not work for drawings using a different layout. The parameters, used for pre-processing, clustering and post-processing could be further optimized. Additionally, the implementation was only tested on Firefox browsers. Security and Privacy aspects as well as performance have not been taken into considerations yet. These issues could also be taken care of in further work. In addition the user interface could be further optimized and adjusted for being used in production. Examples for these potential adjustments:

- Giving the user control of the EPS value and therefore influencing the clustering result.
- Additional analysis of which parameters influence the optimal EPS value could be done.
- The user could mark which elements have been extracted correctly and which are not. This could then be used to further optimize the extraction process or to automatically adjust the parameters for the respective drawing.
- Storing the last input for a specific drawing and using these inputs as placeholder for the next user.
- Important elements can be marked explicitly in the drawing to facilitate the process of distinguishing between essential and non-essential information.
- The input column ”Relevant”, which is currently just used for additionally input, could also be used for the discrimination between important and additional information by letting an expert do the first measuring. The information which elements are essential is then stored and only these are shown to future users (measuring the same workpiece).

- If a clear distinction between essential and non-information is possible, the elements shown in the user interface could be shown accordingly, ranging from the most important values first to the least important ones last.
- The highlighting could also be adjusted to highlight most important values differently.
- As mentioned in the expert interview, the prototype could also be adapted to be used as quality control for the drawings itself, this would require adapting the user interface as well.

This thesis focused on extracting information via clustering and regular expressions, so there is still potential using other algorithms. Using neural networks to determine the respective measurements and tolerances could be interesting as well as using another input format.

Appendix A

Source Code

The prototype is, as mentioned in chapter 4 divided into two separate implementations, the extraction tool and the web application. The source code is available at a GitHub repository. For both implementations the required dependencies can either be downloaded using ”pip” or everything can be installed at once using ”pipenv”, which automatically creates a virtual environment and downloads all dependencies. To test the service, two drawings, which are also part of the evaluation, are provided at the repository as well. [1]

Software required by both implementations:

- python, version: 3.7
- redis, version: 3.3.11 (downloaded via pip3)
- pipenv, version: 3.3.11 (downloaded via pip3), optional

A.1 Build Extraction Tool

Link to repository:
`https://github.com/bscheibel/masterthesis_extraction.git`

Requirements:

- source code cloned from repository

[1] https://github.com/bscheibel/masterthesis_extraction/blob/master/Laeufer.PDF by Peter Travnicek https://github.com/bscheibel/masterthesis_extraction/blob/master/Stahl_Adapterplatte.PDF by Marcel Fuschelberger

- beautifulsoup4, version: 4.8.0 (downloaded via pip3)
- numpy, version: 1.17.4 (downloaded via pip3)
- pandas, version: 0.25.3 (downloaded via pip3)
- sklearn, version: 0.25.3 (downloaded via pip3)

A.1.1 Adapting the paths and setting the parameters

For the extraction tool the paths of the project directory have to be set in the "main.py" file under the variable "path". This path is then used in all functions where results are temporarily stored and used again. As input parameters the "uuid" is needed, which is set automatically as a new PDF is uploaded to the web application. Additionally, the name and path to the PDF file, the database parameters, and, optional, a custom EPS value for the clustering process can be set.

If all requirements are met, the application can be run from the command line, or called by another application, by using the command "python3 main.py" with the needed input parameters.

A.2 Build Web Service

Link to repository:
`https://github.com/bscheibel/masterthesis_extraction.git`

Requirements:

- source code cloned from repository
- flask, version: 1.1.1 (downloaded via pip3)
- pypdf2, version: 1.26.0 (downloaded via pip3)
- gunicorn, version: 20.0.2 (downloaded via pip3), optional, for production deployment with nginx server

A.2.1 Adapting the paths and setting the parameters

For the web application, four paths have to be set. Both of them are in the "views.py" file. The first path is called "path" and refers to the project directory. The other one is called ”path_extraction” and refers to the directory of the extraction tool, as this tool is called by the web application. The path where the image, which is used for vizualisation in the browser, is set in the variable ”path_image”. Lastly, the configuration for ”redis” has to be set in ”db_params”. Additionally, the regulatory documents that can be provided have to be put into the folder ”app/static/isos”.

Otherwise, no additional information can be provided, as regulatory documents are copyright protected and can not be put publicly on the repository. No input parameters for the application itself are needed, as the user will input the PDF file using the browser.

A.3 Build via pipenv

If ”pipenv” is used, just got to the directory of th implementation and the command ”pipenv install”, automatically creates a virtual environment and installs all dependencies. To run the software, type in the command ”pipenv run python” plus the name of the application e.g. ”main.py” or in case of running the flask application ”pipenv run flask run”. However, running the application via ”pipenv” only works if your are in the same directory as the application.

If all these requirements are met, the web application can be run by entering ”flask run” at the command line while being in the project directory.

Bibliography

[1] A. Kusiak, "International Journal of Production Research Smart manufacturing Smart manufacturing," *International Journal of Production Research*, vol. 7543, no. October, pp. 1–10, 2017.

[2] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0: State of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[3] T. Lu, Y. Yang, R. Yang, and S. Cai, "Knowledge extraction from structured engineering drawings," *Proceedings - 5th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2008*, vol. 2, pp. 415–419, 2008.

[4] T. C. Henderson, *Analysis of engineering drawings and raster map images.* Springer New York, jan 2014.

[5] S. Labisch and C. Weber, *Technisches Zeichnen Selbstständig lernen und effektiv üben.* Viewegs Fachbücher der Technik, 3 ed., 2008.

[6] C. F. Moreno-García, E. Elyan, and C. Jayne, "New trends on digitisation of complex engineering drawings," *Neural Computing and Applications*, vol. 1, pp. 1–18, jun 2018.

[7] D. Dori and Y. Velkovitch, "Segmentation and Recognition of Dimensioning Text from Engineering Drawings," *Computer Vision and Image Understanding*, vol. 69, no. 2, pp. 196–201, 1998.

[8] F. L. Krause, H. Jansen, G. Großmann, and G. Spur, "Automatic Scanning and Interpretation of Engineering Drawings for CAD-Processes," *CIRP Annals - Manufacturing Technology*, vol. 38, no. 1, pp. 437–441, 1989.

[9] "MBE PMI Validation and Conformance Testing Project — NIST." https://www.nist.gov/el/systems-integration-division-73400/mbe-pmi-validation-and-conformance-testing-project.

[10] A. Habed and B. Boufama, "Dimension sets detection in technical drawings," in *Vision Interface*, 1999.

[11] S. Vaishnavi, V., Kuechler, W., and Petter, "Design Science Research in Information Systems," 2012.

[12] B. Kuechler and V. Vaishnavi, "On theory development in design science research: Anatomy of a research project," *European Journal of Information Systems*, vol. 17, no. 5, pp. 489–504, 2008.

[13] U. Kurz and H. Wittel, *Böttcher / Forberg Technisches Zeichnen.* Vieweg + Teubner, 2011.

[14] B. Bielefeld and I. Skiba, *Basics Fundamentals of Presentation Technical Drawing.* Birkhäuser, 2017.

[15] D. Smith, A. Ramirez, and A. Fuller, *TECHNICAL DRAWING 101 with AutoCAD 2017.* SDC Publications, 6th ed., 2016.

[16] U. Kurz and H. Wittel, *Konstruktives Zeichnen Maschinenbau.* Springer Vieweg, 2017.

[17] "FreeCAD: Your own 3D parametric modeler." https://www.freecadweb.org/.

[18] R. Grishman, "Information Extraction," *IEEE Intelligent Systems*, vol. 30, pp. 8–15, 2015.

[19] M. Mulins, "Information extraction in text mining," *Computer Science Graduate and Undergraduate Student Scholarship*, 2008.

[20] O. Maimon and L. Rokach, eds., *Data Mining and Knowledge Discovery Handbook.* Springer, 2 ed., 2005.

[21] G. Miner, J. Elder, A. Fast, T. Hill, R. Nisbet, D. Delen, and B. Spencer, *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications.* Elsevier Inc., 2012.

[22] C. Archibald, P. Kwok, and C. Gros, "Automatic Understanding of Technical Drawings: Symbol Extraction by Geometric and Morphological Methods," *Research in Computer and Robot Vision*, pp. 347–366, 1995.

[23] P. M. Devaux, D. B. Lysak, and R. Kasturi, "A complete system for the intelligent interpretation of engineering drawings," *International Journal on Document Analysis and Recognition*, vol. 2, no. 2-3, pp. 120–131, 1999.

[24] P. Vaxivière and K. Tombre, "Knowledge organization and interpretation process in engineering drawing interpretation," *Proc. IAPR Workshop on Document Analysis*, pp. 313–321, 1994.

[25] P. Vaxivière and K. Tombre, "Celesstin: CAD Conversion of Mechanical Drawings," *Computer*, vol. 25, no. 7, pp. 46–54, 1992.

[26] S. Ablameyko, V. Bereishik, O. Frantskevich, M. Homenko, and N. Paramonova, "A system for automatic recognition of engineering drawing entities," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, vol. 2, pp. 1157–1159, IEEE Comput. Soc, 2002.

[27] M. J. Fonseca, A. Ferreira, and J. A. Jorge, "Content-based retrieval of technical drawings," *International Journal of Computer Applications in Technology*, vol. 23, no. 2/3/4, p. 86, 2005.

[28] D. R. Kasimov, A. V. Kuchuganov, and V. N. Kuchuganov, "Individual strategies in the tasks of graphical retrieval of technical drawings," *Journal of Visual Languages and Computing*, vol. 28, pp. 134–146, 2015.

[29] B. T. Messmer and H. Bunke, "Automatic learning and recognition of graphical symbols in engineering drawings," in *Graphics Recognition Methods and Applications. GREC 1995. Lecture Notes in Computer Science* (K. R. and T. K., eds.), (Berlin, Heidelberg), pp. 123–134, Springer, 1996.

[30] X. L. Hoang, E. Arroyo, and A. Fay, "Automatische Analyse und Erkennung graphischer Inhalte von SVG-basierten Engineering-Dokumenten," *At-Automatisierungstechnik*, vol. 64, no. 2, pp. 133–146, 2016.

[31] R. Rahul, S. Paliwal, M. Sharma, and L. Vig, "Automatic Information Extraction from Piping and Instrumentation Diagrams," *CoRR*, vol. abs/1901.1, 2019.

[32] H. Zhang and X. Li, "Data Extraction from DXF File and Visual Display," in *HCI International 2014 - Posters' Extended Abstracts: International Conference, HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings, Part I* (C. Stephanidis, ed.), vol. 434, pp. 286–291, 2014.

[33] Z. Sukimin and H. Haron, "Geometric entities information for feature extraction of solid model based on DXF file," in *Proceedings - International Symposium on Information Technology 2008, ITSim*, vol. 4, 2008.

[34] B. S. Prabhu, "Automatic extraction of manufacturable features from CADD models using syntactic pattern recognition techniques," *International Journal of Production Research*, vol. 37, no. 6, pp. 1259–1281, 2002.

[35] B. Ye, J. Liu, B. Wu, and C. Wu, "New method of feature recognition from engineering drawings based on multi-granularity information acquisition," *6th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009*, vol. 5, pp. 129–133, 2009.

[36] Y. Wang and M. Liang, "A new proposal of data extraction from DXF graphics in power system," in *Proceedings - 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2010*, vol. 3, pp. 839–842, 2010.

[37] I. Kumari and A. M. Magar, "Dxf file extraction and feature recognition," *International Journal of Engineering and Technology*, vol. 4, no. 2, pp. 93–96, 2012.

[38] Zhaoyang Lu, "Detection of text regions from digital engineering drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 431–439, aug 2002.

[39] Y. Cao, H. Li, and Y. Liang, "Using engineering drawing interpretation for automatic detection of version information in CADD engineering drawing," *Automation in Construction*, vol. 14, no. 3, pp. 361–367, 2005.

[40] R. Sulaiman, M. Fahmi Mohamad Amran, and N. Amlya Abd Majid, "A Study on Information Extraction Method of Engineering Drawing Tables," *International Journal of Computer Applications*, vol. 50, no. 16, pp. 43–47, 2012.

[41] Z. Jiang and X. Feng, "An Information Extraction of Title Panel in Engineering Drawings and Automatic Generation System of Three Statistical Tables A . The cell description of BOM information table The data corifiguration of BOM iriformation," in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 1, pp. V1–297–V1–301, IEEE, 2010.

[42] J. Zhang, L. Zhao, and Y. Hao, "Multi-level block information extraction in engineering drawings based on depth-first algorithm," in *Advanced Materials Research*, vol. 468-471, pp. 2100–2103, 2012.

[43] C. J. Romanowski, R. Nagi, and M. Sudit, "Data mining in an engineering design environment: Or applications from graph matching," *Computers and Operations Research*, vol. 33, no. 11, pp. 3150–3160, 2006.

[44] P. Banerjee, S. Choudhary, S. Das, H. Majumdar, R. Roy, and B. B. Chaudhuri, "Automatic Hyperlinking of Engineering Drawing Documents," *Proceedings - 12th IAPR International Workshop on Document Analysis Systems, DAS 2016*, pp. 102–107, 2016.

[45] M. Ondrejcek, J. Kastner, R. Kooper, and P. Bajcsy, "Information Extraction from Scanned Engineering Drawings," *Aperture*, no. Twr 841, pp. 1–29, 2009.

[46] S. Das, P. Banerjee, B. Seraogi, H. Majumder, S. Mukkamala, R. Roy, and B. B. Chaudhuri, "Hand-written and machine-printed text classification in architecture, engineering and construction documents," *Proceedings of International Conference*

on Frontiers in Handwriting Recognition, ICFHR, vol. 2018-Augus, pp. 546–551, 2018.

[47] A. K. Das and N. A. Langrana, "Recognition of Dimension Sets and Integration with Vectorized Engineering Drawings," *Computer Vision and Image Understanding*, vol. 68, no. 1, pp. 90–108, 1997.

[48] C. P. Lai and R. Kasturi, "Detection of Dimension Sets in Engineering Drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 848–855, 1994.

[49] B. S. Prabhu, S. Biswas, and S. S. Pande, "Intelligent system for extraction of product data from CADD models," *Computers in Industry*, vol. 44, no. 1, pp. 79–95, 2001.

[50] W. .-. Masing, T. . Pfeifer, and R. Schmitt, *Masing Handbuch Qualitaetsmanagement.* Hanser, 2014.

[51] "dxfgrabber · PyPI." https://pypi.org/project/dxfgrabber/.

[52] Autodesk, "DXF Reference," *AutoCAD 2008*, no. February 2009, p. 306, 2007.

[53] "STEP for Geometric Dimensioning and Tolerancing." https://www.steptools.com/stds/step/step_3.html.

[54] "Learning Center: Types of PDFs - Image-Only, True PDF, Searchable PDF." https://www.abbyy.com/en-au/finereader/pdf-types/.

[55] S. Pitale and T. Sharma, "Information Extraction Tools for Portable Document Format," *International Journal of Computer Technology Application*, vol. 2, no. 6, pp. 2047–2051, 2011.

[56] "XpdfReader." http://www.xpdfreader.com/.

[57] "PyPDF2 Documentation — PyPDF2 1.26.0 documentation." https://pythonhosted.org/PyPDF2/#.

[58] "tika-python." https://github.com/chrismattmann/tika-python.

[59] "textract — textract 1.6.1 documentation." https://textract.readthedocs.io/en/latest/.

[60] "OCR - Optical Character Recognition erklärt — Learncenter." https://www.abbyy.com/de-de/finereader/what-is-ocr/.

[61] "Optical Character Recognition (OCR) - How it works." https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/.

[62] "pytesseract · PyPI." https://pypi.org/project/pytesseract/.

[63] "Python RegEx." https://www.w3schools.com/python/python_regex.asp.

[64] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Annals of Data Science*, vol. 2, pp. 165–193, jun 2015.

[65] DIN, "DIN ISO 2768 Allgemeintoleranzen," 1991.

[66] K. Winter and S. Rinderle-Ma, "Untangling the GDPR Using ConRelMiner," *CoRR*, vol. abs/1811.0, pp. 1–7, 2018.

[67] "Camelot: PDF Table Extraction for Humans — Camelot 0.7.3 documentation." https://camelot-py.readthedocs.io/en/master/.

[68] W. T. Adrian, N. Leone, and M. Manna, "Ontology-driven Information Extraction," *CoRR*, vol. abs/1512.0, 2015.

[69] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle, "Ontology-based extraction and structuring of information from data-rich unstructured documents," in *CIKM*, pp. 52–59, 1998.

[70] "textract · PyPI." https://pypi.org/project/textract/.

[71] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, 2017.

[72] "scikit-learn: machine learning in Python — scikit-learn 0.21.3 documentation." https://scikit-learn.org/stable/.

[73] "Redis." https://redis.io/.

[74] "Flask — The Pallets Projects." https://www.palletsprojects.com/p/flask/.

[75] K. M. Ting, *Confusion Matrix*, p. 209. Boston, MA: Springer US, 2010.

[76] K. M. Ting, *Precision and Recall*, p. 781. Boston, MA: Springer US, 2010.