Towards Automation of Regulatory Compliance Checking in the Product Design Phase

Malte Ramonat[1], Andreas W. Müller[2] and Alexander Fay[1]
[1]Institute of Automation, Helmut Schmidt University Hamburg, Holstenhofweg 85, Hamburg, Germany
[2]Data & Analytics Governance, Schaeffler AG, Herzogenaurach, Germany

Keywords: Regulatory Compliance Checking, Table Extraction, Formula Extraction, Ontology Design.

Abstract: The process of checking if a designed product is compliant with standards is time-consuming and error-prone. This paper presents an approach for the automation of compliance checking using tables and formulae of standards as information sources. An ontology is created to enable comparisons between parameter values specified in standards in the form of a PDF document and parameter values of a designed product saved in a 3D PDF document. The extraction of regulatory information from PDF documents is discussed and software tools for information extraction are compared.

1 INTRODUCTION

Technical standards provide knowledge for maintaining a certain quality level for products or services. Especially during the product development process, standards specify a leeway for the product design. The Portable Document Format (PDF) has become the most common digital format in which standards are provided by standards committees. PDF documents are widely used throughout the product life cycle due to their portability and platform independence. Information within PDF documents is either unstructured or semi-structured, i.e. it is human-readable but not interpretable for machines (Khusro et al., 2015).

This lack of machine-interpretability makes the process of verifying the compliance with requirements from regularities such as technical standards labor-intensive. In the process of regulatory compliance checking (RCC) during the product design phase, the engineer first needs to find the standards relevant for the designed product. Then, the engineer compares a required value from a specific section of a standard to a value of the designed product and possibly makes adjustments to the product design. This process of RCC has to be repeated multiple times during the product design phase. Thus, manual RCC is time-intensive and error-prone (Manoharan, 2019). If technical dependencies between different engineering departments occur for the designed product, errors in RCC of one department can have a big impact on the whole development process (Jager, 2011). Moreover, RCC is not necessarily only conducted by the company designing the product but also by certification organisations. In the certification process multiple feedback loops between certification organisation and the manufacturer are necessary if the designed product is not fit for certification. Each such feedback loop increases the manual effort for RCC for both involved parties. Due to the amount of manual labor put into RCC, both manufacturers and certification organisations depend significantly on the regularity knowledge of their involved employees. An automation of the RCC process is highly desirable, as it would not only accelerate the design and certification of new products but also reduce the risk of errors during the process and decrease the dependency of a company on the knowledge of single employees. In order to automate the RCC, standard documents need to be made machine-interpretable. Ontologies can be used to enable machine-interpretability of requirements stated in the standards. An approach is presented in this paper demonstrating how ontologies can be connected to requirements stated in standard PDF documents.

The remainder of this paper is organized as follows: In Section 2 a scenario for the application is presented. Section 3 shows an analysis of publications related to automation of RCC. In Section 4 the proposed approach is outlined with its main parts being described in Subsections 4.1 and 4.2. Section 5 completes the paper with a conclusion and outlook.

Ramonat, M., Müller, A. and Fay, A.
Towards Automation of Regulatory Compliance Checking in the Product Design Phase.
DOI: 10.5220/0010644500003064
In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - Volume 2: KEOD*, pages 136-143
ISBN: 978-989-758-533-3; ISSN: 2184-3228

2 APPLICATION SCENARIO

It is common practice for engineers in product design to reuse existing concepts, e.g. in form of 3D CAD models, to create complex designs. Depending on the respective product and its designated fields of use, such supplied models must meet various standards-based quality criteria. In a typical workflow an engineer needs to know all the relevant criteria before he or she can choose to apply a reusable concept in a particular context. Alternatively, he or she must look up, find and correctly interpret the necessary information in the potentially very large knowledge corpus of the relevant standards documents.

In the example shown in Figure 1 the engineer needs to check whether the dimensioning and surface properties of the part described in the 3D PDF file are compliant with underlying standards.

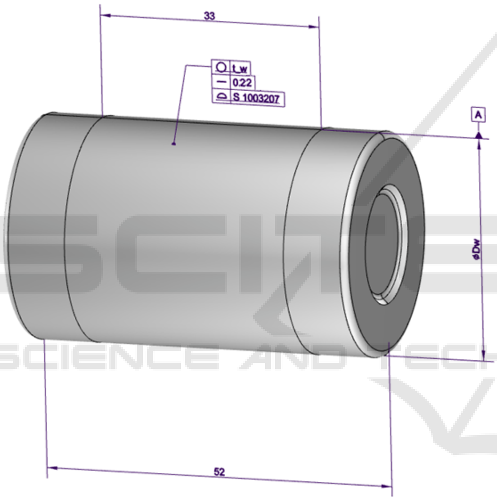
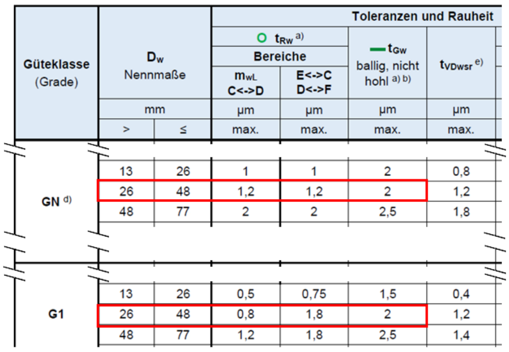


Figure 1: Example for a 3D PDF with sample properties.

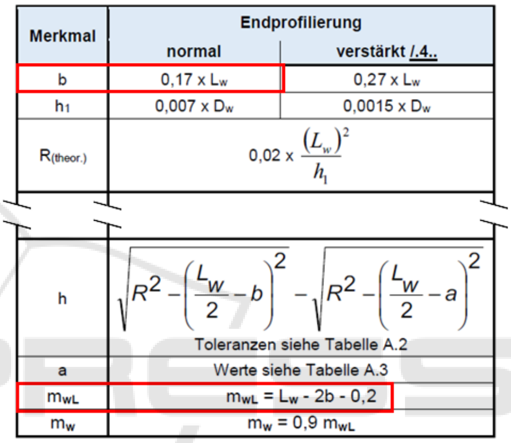
The engineer can currently only gather the information necessary for checking the compliance with these standards from PDF documents. In the referenced standards the necessary dimensioning and surface information are also given in human-readable form of tables and formulae. Hence, the engineer has to read and understand the table shown in Figure 2, and may even have to further perform calculations according to the applicable formulae (see Figure 3). In all cases the relevant attributes and their values must be properly identified in and taken from the 3D PDF product design and matched with the correct attributes in the correct tables and formulae.

In order to provide the engineer with a time-saving automatic checking of these criteria by means of an RCC pipeline, the involved values, tables, and formulae must all be semantically well-described.



Güteklasse (Grade)	D_w Nennmaße		Toleranzen und Rauheit			
			○ t_{Rw} a) Bereiche		— t_{Gw} ballig, nicht hohl a) b)	t_{VDwsr} e)
			m_{wL} C<->D	E<->C D<->F		
	mm		µm	µm	µm	µm
	>	≤	max.	max.	max.	max.
GN d)	13	26	1	1	2	0,8
	26	48	1,2	1,2	2	1,2
	48	77	2	2	2,5	1,8
G1	13	26	0,5	0,75	1,5	0,4
	26	48	0,8	1,8	2	1,2
	48	77	1,2	1,8	2,5	1,4

Figure 2: Example section of a dimensioning table with complex header structures and sample values.



Merkmal	Endprofilierung	
	normal	verstärkt /.4..
b	0,17 x L_w	0,27 x L_w
h_1	0,007 x D_w	0,0015 x D_w
$R_{(theor.)}$	0,02 x $\frac{(L_w)^2}{h_1}$	
h	$\sqrt{R^2 - \left(\frac{L_w}{2} - b\right)^2} - \sqrt{R^2 - \left(\frac{L_w}{2} - a\right)^2}$ Toleranzen siehe Tabelle A.2	
a	Werte siehe Tabelle A.3	
m_{wL}	m_{wL} = L_w - 2b - 0,2	
m_w	m_w = 0,9 m_{wL}	

Figure 3: Example section of a profiles table with complex cell contents and sample values.

3 RELATED WORK

Various efforts have been made towards the automation of the RCC process. (Zhang and El-Gohary, 2015) propose a method for automatic RCC in the construction sector by using ontology-based logic clauses. Requirements are automatically extracted from a PDF document using natural language processing. The extracted information is formalised into logic clauses that can be used for reasoning. The publication is followed by (Zhang and El-Gohary, 2017) in which a reasoning schema for RCC is presented. Both publications focus on textual requirements.

In (Beach, 2013) a framework for RCC in the construction sector is described, in which domain experts create and maintain requirements from standards. Requirements are extracted from regulation documents and are enriched with metadata by domain experts. The requirements are then transferred into a rule engine, which can be used for RCC. In this publication textual requirements are focused. Due to the manual extraction and maintenance of requirements the proposed framework is labor-intensive.

In (Manoharan, 2019) the digitisation of requirements from standards and their integration into the product design process is examined. Information from standards is manually extracted and uploaded to a graph database, which can be accessed by CAD tools. For the presented method the requirements in the standard had to be manually extracted and converted into the JSON format from the standard. The authors criticise the manual effort for the information extraction and upload into the database and advise against using this method in general.

In a following publication (Loibl et al., 2020) propose a procedure for the transformation of standards' contents into a machine-executable form. Standards are classified according to their eligibility for a conversion into a machine-executable format. It is stated that standards primarily containing tables, formulae and text in form of property specifications are especially suited for machine-executability because they possess a high level of unambiguousness. To validate their approach, a data-graph from the standard is transformed into a table which is formalized into a machine-executable format. The information is then transferred to a graph database and provided by webservices.The steps described above are carried out manually making this approach labor-intensive. Formulae and tables in the original PDF document are not addressed.

The majority of previous efforts discuss the automation of the RCC process for the construction sector. Automated RCC for the product design phase should be studied more profoundly due to possible time savings and error prevention. In previous publications mostly requirements in textual form were focused. Tables and formulae are less ambiguous than textual requirements and should therefore be utilised in the automation of RCC. To the best of our knowledge no authors have discussed an automation of RCC for the product design phase using tables and formula as the source of requirements from standards.

4 APPROACH TO AUTOMATIC COMPLIANCE CHECKING

The automation of the RCC process can be achieved in two different ways. The first way is to create standards in a machine-interpretable format, e.g. in the machine-readable Extensible Markup Language (XML) format. This is state-of-the-art in the standardisation process. In such standard documents, tags are used to add semantics to different sections of the document. However, the tags used are not detailed enough to make the standard machine-interpretable (Loibl et al., 2020). Currently, efforts are taken to change the creation process of standards to allow that, in future, machine-interpretable standards are created. However, the timescale until these efforts will be implemented is unclear. This is due to the complex development process of a semantic tag set and also because of the amount of manual effort required for the semantic tagging of standards' contents. It is also unclear if the envisioned tag set will grant a suitable level of accuracy for companies as the responsibility of tag set creation lies with standards committees.

The second way for the automation of RCC is a post-processing of the existing standards, which have been provided as PDF documents. Following this way, results can be obtained faster because less coordination between organisations is necessary. Additionally this post-processing can be applied to all existing standards, whereas the first way can only be applied to newly created standards. A post-processing method could also be applied more easily to internal standards of companies.

In the following sections, the second way is being followed, i.e. a post-processing approach to enable automatic RCC using PDF files of standards is proposed. Thus, information is extracted especially from tables and formulae of standards in form of PDF files by means of appropriate software tools. The extracted information is mapped into an ontology. The information from the product description in form of a 3D PDF file is also extracted and mapped into the ontology. Thus, within the ontology, compliance checks can be executed.

4.1 Information Model for Regulatory Compliance Checking

In order to check if the properties of the product description match the required values in a standard, both values have to be made comparable. This can be achieved by loading the information from both sources into the same information model (IM). In subsection 4.1.1 a suitable type of IM is chosen based on established requirements. A modeling language for the chosen type of IM is selected. In subsection 4.1.2 a prototypical implementation for the IM is described.

4.1.1 Information Model Type and Modeling Language Selection

The type of IM used for implementation has to meet certain requirements. These IM requirements (IMR) are derived from the application scenario and from a literature review. They are described hereinafter in order of their importance to the application scenario.

IMR1 (*Machine-interpretability*): In order to automate RCC, the IM needs to be machine-interpretable. To achieve this, the IM should store information and also provide means to enrich it with contextual meaning, i.e. semantics (Bettini, 2010)

IMR2 (*Value comparison*): The IM must provide means for the automatic comparison of values stated in tables or formulae within a standard to parameter values presented in the product description. Semantic structures of tables and formulae as well as means to make both values comparable must be supported to achieve this.

IMR3 (*Compatibility*): The IM must be accessible by programs so that compliance checks can be applied automatically. For this purpose, the IM must provide non-proprietary interfaces so that other programs can query the database from outside.

IMR4 (*General applicability*): In order for the IM to be be applicable to other application scenarios, it needs to be reusable and extendable (Glawe, M., et al., 2015).

For the implementation of the IM, relational databases and ontologies are compared. Both relational databases and ontologies enable value comparision and can be accessed by other programs. Machine-interpretability is only supported by ontologies because their semantic triple structure allows the modelling of complex semantics. Additionally, ontologies provide more flexibility compared to relational databases as they can easily be extended and reused (Loibl et al., 2020). Therefore, ontologies have been chosen for the implementation.

The Web Ontology Language (OWL) is suitable to model an ontology for the application scenario due to its high level of formalisation and because of its compatibility to various software tools. Also, an ontology built with OWL can easily be connected to other ontologies due to the open world assumption of OWL.

4.1.2 Information Model Formalisation

An ontology has been created using OWL to formalise information from both tables and formulae. Their structures have been represented such that the information extracted from standards can be mapped to the ontology. An excerpt of the ontology is depicted in Figure 4 reflecting the structure of the table in Figure 2.

Firstly, terminological components, i.e. T-Box elements, are built to formalise the general structure of tables. Classes are depicted in white. Object properties and datatype properties are shown in blue and red, respectively. The table structure is reflected by classes for columns, subcolumns, rows and subrows as well as for data-cells and header-cells. Secondly, assertional components, i.e. A-Box elements, are added. Individuals are created for columns and subcolumns as well as for rows and subrows, they are depicted in purple. They are connected to cell-entry-individuals, which hold parameter values as datatype properties. Individuals for header-cells holding parameter names as datatype properties are also connected to the cell-entry-individuals. This way, names and values of parameters are connected. Names and values of parameters of the product description in the 3D PDF file are included in the ontology as datatype properties of 3D-PDF-parameter-individuals. Means of comparison are provided by linking the individuals of header-cells and 3D-PDF-parameters to individuals of the class ”Parameter”. The individuals of the three different classes share the same value for the datatype property ”hasName”. A comparison between table values and 3D PDF values can thus be conducted by SPARQL queries. In order for the value comparison to work, it is important that the same naming convention is followed for parameter names of both the standard and the product description (Hildebrandt, 2020). If this is not the case, a workaround using matching tables has to be used.

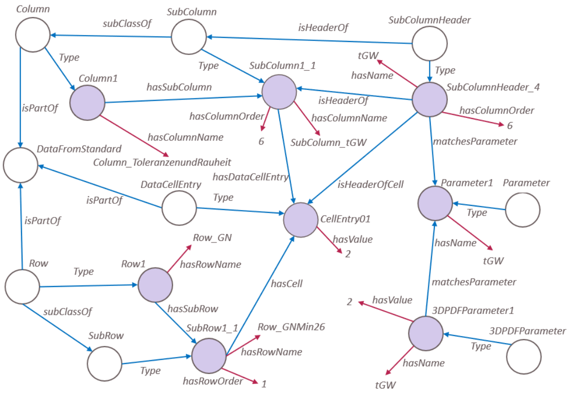


Figure 4: Ontology for allocation of parameter values from tables.

4.2 Extraction of Information from Standards

In order to transfer the required values stated in tables and formulae of standards into the created ontology, they have to be extracted by means of a software tool. In Subsection 4.2.1 different software tools for content extraction are compared based on established requirements. After a suitable software has been chosen, its output is analysed in Subsection 4.2.2.

4.2.1 Selection of Extraction Software

The extraction software has to meet certain requirements to be suitable for the application scenario. These software requirements (SR) described hereinafter result from a literature review and an analysis of the application scenario.

SR1 (*Recognition and extraction of content*): Preceding the extraction, the relevant information in the standard has to be identified. Tables and formulae have to be recognised. Contents of tables and formulae such as numerical values and formula symbols have to be recognised. For table recognition the content of individual table cells should be distinguished (Khusro et al., 2015). The distinction between header- and data-cells is also desirable because header-cells can add semantics to data-cells of the same column (Yildiz et al., 2005). Formulae should be recognised, even if their shape is changed due to mathematical operators like fractions or sums. Indexes and exponents of formula elements as well as mathematical operators should be recognised (Chan and Yeung, 2000).

In addition to the recognition of table and formula elements they should also be extracted.

SR2 (*High quality results*): The recognition of tables and formula should yield high quality results. Precision and recall can be used for result evaluation (Wei et al., 2006). Precision is defined as the number of correctly recognised objects in relation to the total number of recognised objects in the document, whereas recall is defined as the number of correctly recognised objects in relation to the total number of objects in the document. Simple and complex tables with connected cells should be recognised and extracted with comparable levels of precision and recall (Yildiz et al., 2005). The same applies for the different kinds of formulae.

SR3 (*Broad range of applicability*): To provide a broad range of applicability, the software needs to be usable for different types of document layouts and should run on various operating systems (Pitale and Sharma, 2011).

SR4 (*Mature and standalone software*): For a smooth extraction of standard content, a standalone software, which is not dependent on results of other software, should be used. Using a mature, off-the-shelf software would be desirable because this reduces the setup time. A mature software is also likely to be less error-prone.

SR5 (*Extraction of metadata*): Metadata such as the name of the document and the author or the issuing date are important for the contextualisation of the standard's content and should thus be extracted (Pitale and Sharma, 2011).

SR6 (*Compatibility to the information model*): In Section 4.1 OWL has been chosen for the formalisation of the ontology. The software should therefore extract the structure and content of tables and formulae and present them in a format which can be mapped into OWL.

SR7 (*Availability*): The software tool needs to be available so that it can be used to implement an extraction.

The requirements SR1, SR4, SR6 and SR7 are classified as critical for the application scenario. In the following comparison, a software tool is chosen that meets all critical requirements and also meets the most non-critical requirements.

Previous efforts and commercial software tools have been searched for in a literature review and online research. The software tools best fitting to the application scenario are listed in Table 1. Each software tool is evaluated regarding the fulfillment of the above-mentioned requirements. A distinction is made between fulfilled [✓], partly fulfilled [(✓)] and unfulfilled [X] requirements as well as insufficient information [?] for evaluation. The availability of the software is rated with [✓] if the software is available free of charge, [(✓)] if it is a commercial software and [X] if it cannot be acquired. The software tools are categorised by tools for both table and formula extraction, only table extraction and only formula extraction and are arranged by their availability. The requirements have been evaluated on the basis of the software capabilities described by the software creators and in related publications such as (Khusro et al., 2015), (Perez-Arriaga et al., 2016) or (Constantin et al., 2013).

It becomes apparent that no extraction software meets all requirements. Of the 21 software tools shown in Table 1, six are not available. Twelve software tools do not meet other critical requirements and thus cannot be used. The software tools SectLabel, ABBYY FineReader and Nitro Pro do not recognise information in tables or formulae with satisfactory accuracy. Pdf2xml detects the information but does not extract it. PDF-Extract, pdf2table, PDFFigures2, pdf-table-extract, TableSeer and iText are dependent on results of other software and therefore are no standalone software. The software tools Sumatra PDF and i2OCR generate output in a format which is not compatible with the created OWL ontology described in Section 4.1.

The critical requirements are met by PDFX, Adobe Acrobat Pro and InftyReader. PDFX is an online tool which converts PDF documents into the XML format. The XML format is suitable for the approach due to its compatibility to OWL and because

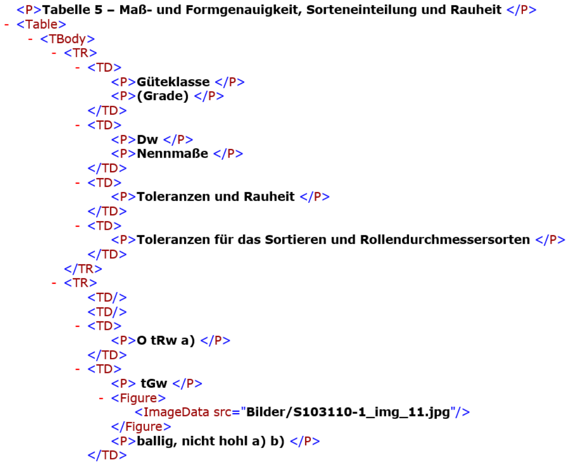
Table 1: Comparison of previous efforts and commercial software for table and formula extraction.

		Requirements						
	Software Tools	SR1	SR2	SR3	SR4	SR5	SR6	SR7
Tables & formulae	PDFX (Constantin et al., 2013)	(✓)	X	✓	(✓)	X	✓	✓
	PDF-Extract (Berg et al., 2012)	(✓)	X	X	X	X	✓	✓
	SectLabel (Luong et al., 2012)	X	?	✓	X	✓	✓	✓
	Adobe Acrobat Pro (Adobe, 2021)	(✓)	X	✓	✓	✓	✓	(✓)
	ABBYY FineReader (ABBYY, 2021)	X	X	✓	✓	X	(✓)	(✓)
	Nitro Pro (Nitro Software, Inc., 2021)	X	X	✓	✓	?	(✓)	(✓)
	(Wei et al., 2006)	(✓)	X	(✓)	✓	X	X	X
Tables	Pdf2table (Yildiz et al., 2005)	✓	X	(✓)	X	X	✓	✓
	PDFFigures2 (Clark and Divvala, 2016)	(✓)	X	X	X	X	X	✓
	pdf-table-extract (Lee et al., 2014)	X	?	X	X	?	✓	✓
	Pdf2xml (Déjean and Meunier, 2006)	X	?	X	✓	X	✓	✓
	TableSeer (Liu, Y., 2009)	✓	?	?	X	✓	✓	✓
	Sumatra PDF (Sumatra, 2021)	(✓)	X	✓	✓	X	X	✓
	iText (iText Group nv, 2021)	(✓)	✓	?	X	X	✓	(✓)
	XONTO (Oro and Ruffolo, 2008)	✓	?	✓	✓	X	✓	X
	PDF-TREX (Oro and Ruffolo, 2009)	✓	?	X	✓	X	✓	X
	TAO (Perez-Arriaga et al., 2016)	✓	?	✓	X	✓	(✓)	X
Formulae	i2OCR (Sciweavers LLC, 2021)	(✓)	?	✓	✓	X	X	✓
	InftyReader (Suzuki, 2004)	✓	(✓)	✓	✓	?	✓	(✓)
	(Garain and Chaudhuri, 2005)	X	?	✓	✓	X	X	X
	MaxTract (Baker et al., 2012)	✓	?	✓	✓	?	(✓)	X

it is both human-readable and machine-interpretable (Schmidberger and Fay, 2007). Because standards are confidential documents of high value, an online conversion tool like PDFX cannot be used due to possible security leakages. The InftyReader meets the critical requirements and is also applicable to different document types. It converts formula content into MathML output. The MathML format is compatible to OWL and enables structuring of mathematical equations in a detailed and hierarchical way (Schmidberger and Fay, 2007). Thus, it is suited for the proposed approach. The InftyReader is limited to the extraction of formulae and does not detect tables. Therefore, formulae within tables cannot be extracted. For many standards such as the one presented in Figure 3, formulae are, however, depicted within tables making the InftyReader not usable. Adobe Acrobat Pro is chosen for both formula and table extraction as it meets all critical requirements. It can be used to convert PDF documents into the XML format which can be mapped into OWL. Adobe Acrobat Pro works for digitally created PDF documents but does not recognise a table or formula if it is added as a figure in the original file. Because of this, Adobe Acrobat Pro does not work for scanned documents. This means that the extraction cannot be generalised to all kinds of PDF documents. It has to be noted that Adobe Acrobat Pro is chosen due to the absence of a satisfactory alternative. Additionally, none of the software tools can distinguish between header- and data-cells of tables. This proves that the technical infrastructure for a fully functional method for post-processing of PDF documents has yet to be implemented.

4.2.2 Analysis of Software Output

Adobe Acrobat Pro has been used to convert the table shown in Figure 2 into the XML format. An excerpt of the output is shown in Figure 5.



```
  <P>Tabelle 5 – Maß- und Formgenauigkeit, Sorteneinteilung und Rauheit </P>
- <Table>
  - <TBody>
    - <TR>
      - <TD>
          <P>Güteklasse </P>
          <P>(Grade) </P>
        </TD>
      - <TD>
          <P>Dw </P>
          <P>Nennmaße </P>
        </TD>
      - <TD>
          <P>Toleranzen und Rauheit </P>
        </TD>
      - <TD>
          <P>Toleranzen für das Sortieren und Rollendurchmessersorten </P>
        </TD>
      </TR>
    - <TR>
        <TD/>
        <TD/>
      - <TD>
          <P>O tRw a) </P>
        </TD>
      - <TD>
          <P> tGw </P>
        - <Figure>
            <ImageData src="Bilder/S103110-1_img_11.jpg"/>
          </Figure>
          <P>ballig, nicht hohl a) b) </P>
        </TD>
```

Figure 5: XML Output of Adobe Acrobat Pro.

Tags are assigned to table rows, columns and to the table itself. header-cells and data-cells are not tagged as such. Multi-row-cells of tables are converted into multiple cells. The content of multi-row-cells is saved in the uppermost cell, all other cells are blank. Multi-column-cells are considered as single cells. Thus, the number of columns per row can vary. This is problematic for the mapping of information from tables with complex header structures such as the table shown in Figure 2 because headers of different rows can in some cases not be assigned to the same column. Formulae are converted into plain text without tags for formula components. Formulae inserted into the original document as a figure are not converted. Because of the lack of tags and the poor quality of the formula output, the mapping of the XML output into the ontology still presents a challenge.

5 CONCLUSION AND OUTLOOK

In this paper it is shown that the automation of RCC is highly desirable because it leads to time savings and less errors. An analysis of the related work shows that the automation of the RCC on the basis of table and formula information has not yet been examined. This research gap has been addressed by this approach, specifically for the comparison of parameter values from a product description to values of a standard. In the approach for automatic RCC an ontology for table and formula information has been formalised. Next, information from a standard and from the product description has been extracted. For this step Adobe Acrobat Pro has been chosen. The XML output of Adobe Acrobat Pro has been analysed with regards for suitability for information mapping into the created ontology.

The use of Adobe Acrobat Pro restricts the approach to the extraction of standards' requirements from digitally created PDF documents. This limits the applicability of the approach. The mapping of the Adobe Acrobat Pro XML output into the ontology presents a challenge because tags for detailed formula and table content are missing. Table header- and data-cells cannot yet be securely transferred into the ontology. Formula calculations still need to be introduced. Previous efforts have shown that there is no easy-to-use-way to implement the calculation of complex formulae into ontologies (Hildebrandt, 2017). Formula calculation needs to be implemented in a different manner.

Future work will focus on the mapping of the XML output into the created ontology. A mapping algorithm needs to be implemented to enable an automation of the RCC process. The created ontology needs to be refined so that it is more generally applicable. Furthermore, an extension of the ontology adding more detailed descriptions to the parameter names is planned. Additional research will be devoted to finding a more suitable extraction software as well as to the integration of formula calculations into the approach.

ACKNOWLEDGEMENTS

This publication resulted from a project between the Schaeffler AG and the Helmut Schmidt University Hamburg. The authors wish to thank the colleagues from Schaeffler Engineering Standards (in alphabetical order: Sören Clodius, Markus Franke, Stefan Gatersleben, Dietmar Lochner, Hans Reichelsdorfer) for providing and discussing application scenarios.

REFERENCES

ABBYY (2021). Finereader pdf. Retrieved March 6, 2021 from https://pdf.abbyy.com/?redirect-from=old-fr-ce.

Adobe (2021). Acrobat pro. Retrieved March 6, 2021 from https://acrobat.adobe.com/us/en/acrobat.html.

Baker, J. B., Sexton, A. P., and Sorge, V. (2012). Maxtract: Converting pdf to latex, mathml and text. In *Intelligent Computer Mathematics*, pages 422–426. Springer Berlin Heidelberg, Berlin, Heidelberg.

Beach, T. H., e. a. (2013). Towards automated compliance checking in the construction industry. In *Database and Expert Systems Applications*, pages 366–380. Springer Berlin Heidelberg, Berlin, Heidelberg.

Berg, Ø. R., Oepen, S., and Read, J. (2012). Towards high-quality text stream extraction from pdf: Technical background to the acl 2012 contributed task. In *50th Annual Meeting of the Association for Computational Linguistics*, pages 98–103, Stroudsburg, PA. Association for Computational Linguistics (ACL). Software available at https://github.com/oyvindberg/PDFExtract.

Bettini, C., e. a. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161–180.

Chan, K. and Yeung, D. (2000). Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition*, 3(1):3–15.

Clark, C. and Divvala, S. (2016). Pdffigures 2.0. In *JCDL'16*, pages 143–152, Piscataway, NJ. IEEE. Software available at https://github.com/allenai/pdffigures2.

Constantin, A., Pettifer, S., and Voronkov, A. (2013). Pdfx fully-automated pdf-to-xml conversion of scientific literature. In *Proceedings of the 2013 ACM symposium on Document engineering - DocEng '13*, page 177, New York, New York, USA. ACM Press. Software available at http://pdfx.cs.man.ac.uk/.

Déjean, H. and Meunier, J. (2006). A system for converting pdf documents into structured xml format. In *Document analysis systems VII*, Lecture notes in computer science, pages 129–140. Springer, Berlin. Software available at https://sourceforge.net/projects/pdf2xml/.

Garain, U. and Chaudhuri, B. B. (2005). A corpus for ocr research on mathematical expressions. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(4):241–259.

Glawe, M., et al. (2015). Knowledge-based engineering of automation systems using ontologies and engineering data. In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015)*, pages 291–300.

Hildebrandt, C., e. a. (2017). Reasoning on engineering knowledge: Applications and desired features. In *European Semantic Web Conference*, volume 10250 of *Lecture notes in computer science*, pages 65–78, Cham. Springer International Publishing.

Hildebrandt, C., e. a. (2020). Ontology building for cyber–physical systems: Application in the manufacturing domain. *IEEE Transactions on Automation Science and Engineering*, 17(3):1266–1282.

iText Group nv (2021). itext. Retrieved March 6, 2021 from https://itextpdf.com/.

Jager, T., e. a. (2011). Mining technical dependencies throughout engineering process knowledge. In *ETFA2011*, pages 1–7. IEEE.

Khusro, S., Latif, A., and Ullah, I. (2015). On methods and tools of table detection, extraction and annotation in pdf documents. *Journal of Information Science*, 41(1):41–57.

Lee, C., Bzdak, J., and Lannon, B. (2014). pdf-table-extract. Retrieved March 6, 2021 from https://github.com/ashima/pdf-table-extract.

Liu, Y. (2009). *Tableseer: Automatic Table Extraction, Search an Understanding*. Dissertation, The Pennsylvania State University. Software available at https://sourceforge.net/projects/tableseer/.

Loibl, A., Manoharan, T., and Nagarajah, A. (2020). Procedure for the transfer of standards into machine-actionability. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 14(2):JAMDSM0022–JAMDSM0022.

Luong, M., Nguyen, T. D., and Kan, M. (2012). Logical structure recovery in scholarly articles with rich document features. In *Multimedia Storage and Retrieval Innovations for Digital Library Systems*, pages 270–292. Software available at https://github.com/knmnyn/ParsCit/tree/master/bin/sectLabel.

Manoharan, T., e. a. (2019). Approach for a machine-interpretable provision of standard contents using welded constructions as an example. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1):2477–2486.

Nitro Software, Inc. (2021). Nitro pro. Retrieved March 6, 2021 from https://www.gonitro.com/.

Oro, E. and Ruffolo, M. (2008). Xonto: An ontology-based system for semantic information extraction from pdf documents. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, pages 118–125. IEEE.

Oro, E. and Ruffolo, M. (2009). Pdf-trex: An approach for recognizing and extracting tables from pdf documents. In *2009 10th International Conference on Document Analysis and Recognition*, pages 906–910. IEEE.

Perez-Arriaga, M. O., Estrada, T., and Abad-Mota, S. (2016). Tao: System for table detection and extraction from pdf documents. *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference*, pages 591–596.

Pitale, S. and Sharma, T. (2011). Information extraction tools for portable document format. *International Journal of Computer Technology 2011*, Vol 2(6):2047–2051.

Schmidberger, T. and Fay, A. (2007). A rule format for industrial plant information reasoning. In *2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007)*, pages 360–367. IEEE.

Sciweavers LLC (2021). i2ocr. Retrieved March 6, 2021 from https://www.i2ocr.com/.

Sumatra (2021). Sumatra pdf reader. Retrieved March 6, 2021 from https://www.sumatrapdfreader.org/free-pdf-reader.

Suzuki, M., e. a. (2004). An integrated ocr software for mathematical documents and its output with accessibility. In *Computers Helping People with Special Needs*, volume 3118 of *Lecture notes in computer science*, pages 648–655. Springer, Berlin and Heidelberg. Software available at http://www.inftyreader.org/.

Wei, X., Croft, B., and McCallum, A. (2006). Table extraction for answer retrieval. *Information Retrieval*, 9(5):589–611.

Yildiz, B., Kaiser, K., and Miksch, S. (2005). pdf2table: A method to extract table information from pdf files. *IICAI*, pages 1773–1785. Software available at http://ieg.ifs.tuwien.ac.at/projects/pdf2table.

Zhang, J. and El-Gohary, N. M. (2015). Automated information transformation for automated regulatory compliance checking in construction. *Journal of Computing in Civil Engineering*, 29(4).

Zhang, J. and El-Gohary, N. M. (2017). Semantic-based logic representation and reasoning for automated regulatory compliance checking. *Journal of Computing in Civil Engineering*, 31(1):04016037.