Automatic Data Sheet Information Extraction for Supporting Model-based Systems Engineering

Kobkaew Opasjumruskit[1][0000−0002−9206−6896],
Sirko Schindler[1][0000−0002−0964−4457], and
Diana Peters[1][0000−0002−5855−2989]

German Aerospace Center, Institute of Data Science,
Mälzerstraße 3, 07745 Jena, Germany
`firstname.lastname@dlr.de`

Abstract. To describe modeling objects in Model-Based Systems Engineering (MBSE) tools, physical properties of these objects are often provided only in data sheets, which are not truly machine-readable. Previously, we proposed a product data hub to exchange spacecraft product information between manufacturers and various MBSE tools. However, issues with heterogeneous structures and semantics of information, such as differences in data format and vocabularies, persist. Using ontologies to maintain product descriptions can mitigate the heterogeneity problem by providing semantic descriptions and supporting different vocabularies for a single concept. To automatically and semantically obtain information from documents that contain tables, lists, and text, we developed an ontology-based information extraction tool. We present how to use the Data Sheets Annotation Tool (DSAT) for, either manually or automatically, extracting information from data sheets, and populating a database with the obtained data. Particularly, we emphasize on the usage of DSAT as a user interface for improving ontologies, which, in turn, are used for a (better) information extraction from the data sheets. Although DSAT is initially created for supporting collaborative systems engineering, it is not limited to the domain of spacecraft design. It can also be applied to other domains, where information needs to be extracted from a multitude of heterogeneous sources.

Keywords: Semantic Technologies for Information-Integrated Collaboration · Ontology for information sharing · Web based cooperation tools

1 Introduction

According to INCOSE [9], MBSE has been created with the vision of using models instead of documents as the formal resource for system engineering activities. It aims on the one hand to integrate different models, and on the other hand to link them with a coherent digital system model, which serves as the single and controlled source of information about the system. These models consist of many parts, each selected out of multiple products provided by different suppliers. The current practice is that engineers manually enter information about components based on the suppliers' PDF data sheets, manually maintained spreadsheets, and their implicit knowledge [11].

To close the gap between these scattered sources of information, in [15], we introduced the idea of a product data hub that enables up-to-date product information to be digitally exchanged between all stakeholders. However, there are heterogeneity issues in terms of data presentation and semantics. For example, the information is sometimes provided in multi-column text with images and graphs, or with tables or lists, and oftentimes mixed. Additionally, the vocabulary used for the same concept can vary substantially, which may lead to ambiguities and requires experts' clarification. Instead of resorting to these manual efforts over and over again, this knowledge can be captured inside a semantic knowledge base. Such a knowledge base needs to be constantly updated to keep up with new developments in the field.

Ontology-Based Information Extraction (OBIE) is one information extraction approach to mitigate these issues as proposed and discussed in [11,17,20]. Most OBIE tools are tailored to extract entities and their relationships but fall short when it comes to extracting literal values, which are the crucial information in the data sheets. They often appear in the form of key-value(-unit) tuples, like weight-200-grams. Furthermore, the vocabulary used in data sheets is highly domain-specific and not consistently used. For example, an attribute of a star sensor, *sun exclusion angle*, is also called *sun angle*, *sun avoidance*, *sun exclusion*, or *sun keep out.* This can be confusing, even for engineers who are familiar with the domain. Missing to detect information can have fatal and costly consequences in the later phases of design and production.

In DSAT-demo [12] we presented a system which provides a human-in-the-loop interface for the automatic extraction of technical properties from data sheets based on an OBIE-pipeline. This paper is a continuation of our aforementioned contribution by providing a graphical user interface. We enhanced the tool DSAT by providing an intuitive user interface so that domain experts, who are not necessarily familiar with ontologies, can suggest the previously undetected attributes or provide corrections to incorrectly identified ones.

2 Related Work

Since most data sheets are provided in PDF format, the text extraction is the first step. Yet, the task is not trivial and there are many solutions available. *PDFminer.six* [14] is an open-source and actively maintained PDF Parser library in Python. It offers an extraction with customizable parameters, such as a page to extract, or coordinations on the page. *Textricator* [18] is an open-source tool to extract data from PDFs and can also handle scanned documents using Optical Character Recognition (OCR), but requires consistently formatted inputs. *Camelot* [4] is and open source software tool to extract tabular data from

PDF files. For an extensive review of tools for extracting data and text from PDFs, please refer to [16].

Text, table layouts, and schematic drawings are equally important in technical data sheets. However, most of the existing tools focus on either text or tables, and, to the extent of our knowledge, there is no unified solution that tackles all of these. To achieve the best result, we use a combination of techniques, i.e. using *PDFminer.six* to extract text, and *Camelot* to detect tables.

Once the raw text is extracted, we use entity recognition tools to detect important keywords and their associated information, particularly, product properties and their values. *Amazon Comprehend* [1] can extract property names and values with their unit, although the relation between them is lost. *DBpedia Spotlight* [6] can detect nouns, pronouns, or specific names with a corresponding entry in DBpedia. However, it extracts words without considering the particular domain, therefore, the results are often associated with unrelated concepts due to similar names. *Intelligent Tagging* [10] extracts names and their surrounding text, which likely contain a corresponding value, but its results tend to be too generalized towards the news domain. *GATE-ANNIE* [8] detects words and recognizes their syntax and category. It is possible to define a custom dictionary for *GATE-ANNIE* to tailor its results to a specific domain.

The aforementioned approaches can detect individual properties, but the customization is not intuitive. Moreover, they detect entities and values separately, so the required relations between them are lost. One approach to detect entities with their relations is Ontology-Based Information Extraction (OBIE).

Baclawski et al. [2] summarize the current trends that combine Machine Learning (ML), information extraction, and ontology techniques to solve complex problems, such as OBIE. Here, the unstructured or semi-structured text is processed using ontologies to extract information. Barkschat [3] exploits technical data sheets to populate ontologies using a classifier model and regular expressions. Likewise, *Smart-dog* [11] extracts data from data sheets of spacecraft parts to populate an ontology. It features an ontology enrichment step but relies extensively on ontology knowledge. Rizvi et al. [17] include irrelevant terms and probably-relevant terms in their ontology to calculate the confidence score of the extracted information.

Based on the lessons learned, we implement OBIE to not only populate and enrich ontologies but also to improve information extraction workflows.

3 System Overview

An overview of our system architecture is shown in Figure 1. The main inputs are data sheets obtained from various sources including websites of manufacturers and retailers. *Continuously Trained Ontologies (ConTrOn)* [13] enables the extraction of information from data sheets and stores the results to a database. The extraction relies on domain-specific ontologies that are constantly adapted using generic, semantic knowledge bases, such as Wikidata [19].

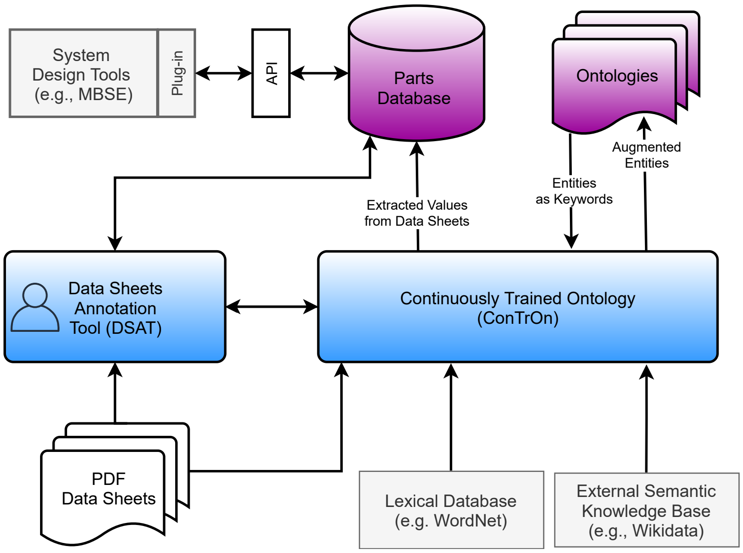
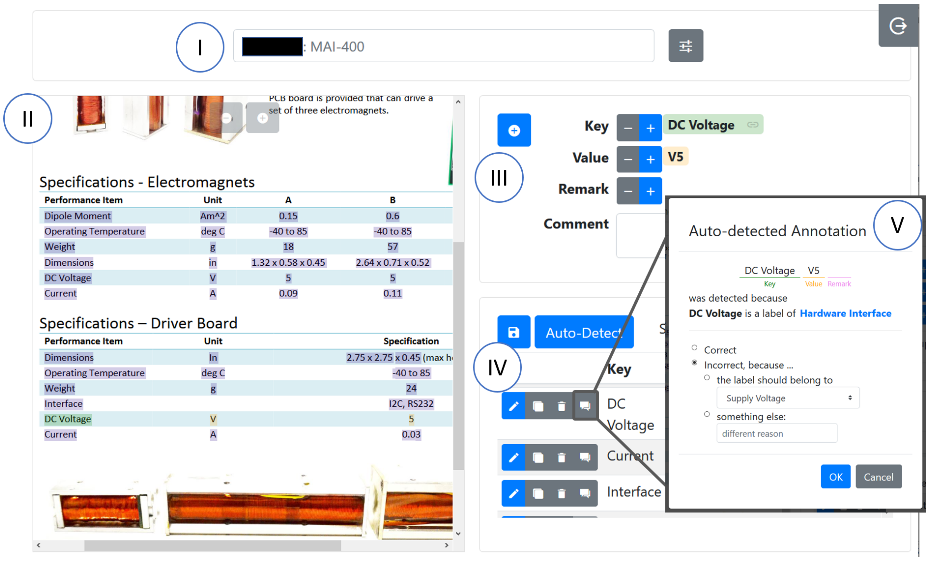

System Design Tools (e.g., MBSE)
Plug-in
API
Parts Database
Ontologies
Augmented Entities
Entities as Keywords
Extracted Values from Data Sheets
Data Sheets Annotation Tool (DSAT)
Continuously Trained Ontology (ConTrOn)
PDF Data Sheets
Lexical Database (e.g. WordNet)
External Semantic Knowledge Base (e.g., Wikidata)

Fig. 1. System overview

The initial, local ontologies were based on satellite component data derived from an MBSE tool (*Virtual Satellite* [7]), European Cooperation for Space Standardization (ECSS), and feedback from domain experts. The ontologies include a core ontology covering concepts common to most satellite parts as well as specialized ontologies specific for the different categories of satellite parts. All ontologies are publicly available from ConTrOn's ontologies [5].

Meanwhile, *Data Sheets Annotation Tool (DSAT)* [12] is not only a standalone tool assisting users for manually annotating data, but is also used as a user interface connecting users to ConTrOn. We use DSAT to review ConTrOn's extracted information and to manually correct any mistake made by the system. Finally, the extracted data is made available to external components or modeling tools, such as MBSE tools via the Parts Database [15].

DSAT displays the information extracted by ConTrOn and allows for user-review before storing them into a database. This web-based application allows users to upload and select their data sheets (see Figure 2-I). Then, the selected data sheet is displayed and essential properties can be highlighted on the left panel (II). Data sheets can be processed by ConTrOn, via the Auto-Detect feature to automatically highlight properties in the data sheet. The highlighted text can be categorized as a key, value, or remark (III). Users can also add custom text in a comment box. One set of a key, a value, a remark, and a comment is denoted as an annotation. All annotations created for the currently selected data sheet are summarized on the bottom right panel (IV). Users can add, edit, or delete the annotations afterwards. The ontologies' classes used for the (automatic) information extraction can be individually reviewed and edited (V). If an attribute (key) is incorrectly identified, e.g. DC Voltage is identified as



[redacted]: MAI-400

PCB board is provided that can drive a set of three electromagnets.

Specifications - Electromagnets

Performance Item	Unit	A	B
Dipole Moment	Am^2	0.15	0.6
Operating Temperature	deg C	-40 to 85	-40 to 85
Weight	g	18	57
Dimensions	in	1.32 x 0.58 x 0.45	2.64 x 0.71 x 0.52
DC Voltage	V	5	5
Current	A	0.09	0.11

Specifications – Driver Board

Performance Item	Unit	Specification
Dimensions	In	2.75 x 2.75 x 0.45 (max h
Operating Temperature	deg C	-40 to 85
Weight	g	24
Interface		I2C, RS232
DC Voltage	V	5
Current	A	0.03

Key: DC Voltage
Value: V5
Remark:
Comment:

Key
DC Voltage
Current
Interface

Auto-detected Annotation

DC Voltage (Key) V5 (Value) (Remark)

was detected because
DC Voltage is a label of **Hardware Interface**

○ Correct
● Incorrect, because ...
○ the label should belong to: Supply Voltage
○ something else: different reason

Fig. 2. Data Sheets Annotation Tool interface allows domain experts to review annotations in a data sheet.

Hardware Interface instead of Supply Voltage, users can suggest the correct description (class), or even suggest a new description. Such feedback from users will be used to update the ontologies in order to assure the correctness of the semantic information. Currently, all the corrections and suggestions must be reviewed by domain experts before applying them to the ontologies. Since we are sharing the ontologies with other systems, the automatic update at this point is not recommended. Eventually, the OBIE process gets improved as well as the quality of information extraction.

4 Conclusion & Future Work

In this paper we presented an integrated system consisting mainly of DSAT and ConTrOn to support engineers in extracting technical information from PDF data sheets. A feedback feature on DSAT enables users to change the key's concept (class) or suggest a new class to our ontologies. The feedback for incorrectly extracted data can be specified either as a class suggestion or as a comment in free-text form. Next, we plan to conduct a user study to evaluate DSAT whether the feedback form is intuitive to use and if the feedback can help improve the quality of the extracted information as well as the ontologies.

References

1. Amazon Comprehend - Natural Language Processing (NLP) and Machine Learning (ML). https://aws.amazon.com/comprehend/, accessed June 25, 2021.

2. Baclawski, K., Bennett, M., Berg-Cross, G., Fritzsche, D. M., Schneider, T., Sharma, R., Sriram, R. D., and Westerinen, A. Ontology Summit 2017 communiqué - AI, learning, reasoning and ontologies. *Applied Ontology 13* (2017), 3–18.
3. Barkschat, K. Semantic information extraction on domain specific data sheets. In *ESWC* (2014).
4. Camelot. https://camelot-py.readthedocs.io/, accessed June 25, 2021.
5. ConTrOn. Contron - spacecraft parts ontology - dsat demo. ”https://zenodo.org/record/4034478”, Sept. 2020.
6. DBpedia Spotlight - Shedding light on the web of documents. https://www.dbpedia-spotlight.org/, accessed June 25, 2021.
7. Fischer, P. M., Lüdtke, D., Lange, C., Roshani, F.-C., Dannemann, F., and Gerndt, A. Implementing Model-Based System Engineering for the Whole Lifecycle of a Spacecraft. *CEAS Space Journal 9*, 3 (July 2017), 351–365.
8. English Named Entity Recognizer. https://cloud.gate.ac.uk/shopfront/displayItem/annie-named-entity-recognizer, accessed June 25, 2021.
9. INCOSE SE Vision 2020. techreport, International Council on Systems Engineering (INCOSE), 2007.
10. Intelligent Tagging & Text Analytics — Refinitiv. https://www.refinitiv.com/en/products/intelligent-tagging-text-analytics, accessed June 25, 2021.
11. Murdaca, F., Berquand, A., Kumar, K., Riccardi, A., Soares, T., Gerené, S., and Brauer, N. Knowledge-based information extraction from datasheets of space parts. In *8th International Systems & Concurrent Engineering for Space Applications Conference* (September 2018).
12. Opasjumruskit, K., Peters, D., and Schindler, S. DSAT: Ontology-based Information Extraction on Technical Data Sheets. In *SEMWEB* (2020).
13. Opasjumruskit, K., Schindler, S., Thiele, L., and Schäfer, P. M. Towards learning from user feedback for ontology-based information extraction. In *Proceedings of the 1st International Workshop on Challenges and Experiences from Data Integration to Knowledge Graphs co-located with the 25th ACM SIGKDD* (2019), vol. 2512 of *CEUR Workshop Proceedings*, CEUR-WS.org.
14. PDFMiner - a tool for extracting information from PDF documents. https://github.com/pdfminer/pdfminer.six, accessed June 25, 2021.
15. Peters, D., Fischer, P. M., Schäfer, P. M., Opasjumruskit, K., and Gerndt, A. Digital availability of product information for collaborative engineering of spacecraft. In *Cooperative Design, Visualization, and Engineering* (Cham, 2019), Y. Luo, Ed., Springer International Publishing, pp. 74–83.
16. Pollock, R. Tools for extracting data and text from pdfs - a review. https://okfnlabs.org/blog/2016/04/19/pdf-tools-extract-text-and-data-from-pdfs.html, Apr. 2016.
17. Rizvi, S. T. R., Mercier, D., Agne, S., Erkel, S., Dengel, A., and Ahmed, S. Ontology-based information extraction from technical documents. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence* (2018), SCITEPRESS - Science and Technology Publications.
18. Textricator. https://textricator.mfj.io/, accessed June 25, 2021.
19. Vrandečić, D., and Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM 57*, 10 (Sept. 2014), 78–85.
20. Wimalasuriya, D. C., and Dou, D. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science 36* (2010), 306–323.