DSAT: Ontology-based Information Extraction on Technical Data Sheets

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

DLR Institute of Data Science
Mälzerstraße 3, 07745 Jena, Germany
{firstname.lastname}@dlr.de

Abstract. Current engineering design processes oftentimes involve transferring information from manufacturer-provided data sheets into domain-specific design tools. While most data sheets are provided only as PDF files, this remains a tedious and manual task. This paper presents the Data Sheets Annotation Tool (DSAT), which assists engineers in gathering the information required in the design process. Using an Ontology-Based Information Extraction (OBIE) method, the properties of components are extracted from data sheets and subsequently presented in an integrated, web-based interface. Engineers can now review and correct these automatic annotations, before exporting them for further use. In the demonstration, we employ a real-world use case rooted in model-based space-system engineering. We show how the automated process can extract relevant component-attributes from technical data sheets and how users can redact the results. We further highlight the impact of content and quality of the underlying ontologies.

1 Introduction

An important, recurring task in many engineering projects is to gather information about components to be used. These are described by their physical properties (e.g., spatial dimensions or mass) and the interfaces they provide or require (e.g., propelling force or power consumption). The acquired descriptions are subsequently fed into domain-specific design tools like Virtual Satellite [3] and allow engineers to compare, combine, and adjust the components based on the project's requirements.

The information required is barely available in machine-processable form, but is usually provided by PDF files (see Fig. 1 for examples). Engineers are required to obtain data sheets of interest, scan these files, and manually copy

Standard NST

Attitude Solution	5 Hz
Sky Coverage	> 99 %
Mass	0.35 kg w/ baffle
Volume	10 x 5.5 x 5 cm
Peak Power	< 1.5W
Field of View	10 x 12 degrees
Sun Keep Out	45 degrees (half cone)
Design Life	> 5 Years (LEO)

Specifications

Performance Item	Specification
Accuracy (Cross Axis / Boresight)	5.7 arcsec / 27 arcsec
Acquisition Time	130 ms Acq, 105 ms Track (typical)
Max Tracking Rate	>2.0°/sec
Update Rate	4 Hz
Star Catalog	Hipparcos
Lens	0.9in f1.2 BK7 Glass
Sun Exclusion w/wo Baffle	45° / 90°
Operating Temperature	-40 to 80 °C
Weight	170g (282 g w/ housing)
Dimensions wo / w case (mm)	50 x 50 x 47 / 55 x 65 x 70
Baffle Dimensions, Wt.	100 x 90 x 195, 135 gm
DC Voltage	5.0 V
Radiation TID (outside of case)	75 krad
Average power consumption	2.0W LIS, 1.5 W Track
Serial Interface	UART TTL / I2C

Fig. 1. An excerpt of data sheets, (left) a star tracker (Standard NST) data sheet from Blue Canyon Tech and (right) a star tracker (MAI-SS) from Adcole Maryland.

the important pieces of information to their respective design tools. Not only is this process tedious and repetitive, it is also error-prone and time consuming.

Existing information extraction approaches can alleviate this task, in particular, Ontology-Based Information Extraction (OBIE) has proven itself a valuable tool to convert text into machine-readable structures as witnessed by [5,7,10]. However, some drawbacks remain for engineering projects. First, most tools are tailored to extract entities and their relationships, but the main information to be extracted in data sheets are key-value(-unit) tuples. Second, the vocabulary used in data sheets is highly domain specific and generally not consistently used. Thus, OBIE approaches that rely on general purpose ontologies are bound to fail here. Finally, incorrectly extracted information is not tolerable. Depending on the scope of the project, an erroneous value for some property can have fatal and very costly consequences later in the development process.

In the following we present the Data Sheets Annotation Tool (DSAT) [6]. It provides a human-in-the-loop interface for the extraction of technical properties from PDF files. Each data sheet is initially processed by an OBIE-pipeline to automatically detect relevant properties of the components described. For this purpose, we employ domain-specific ontologies [2] targeted at the specific types of possible components. The results are presented to engineers who can review, amend to, or remove the automatically created annotations directly on the PDF file, before exporting them to their respective design tool.

2 Data Sheets Annotation Tool (DSAT)

DSAT is a web-based application which displays the data sheets and allows engineers to annotate attributes that will be used by different tools[1]. As shown in

[1] A link to demo video:
https://zenodo.org/record/4034478/files/DSAT-screen-record.mp4

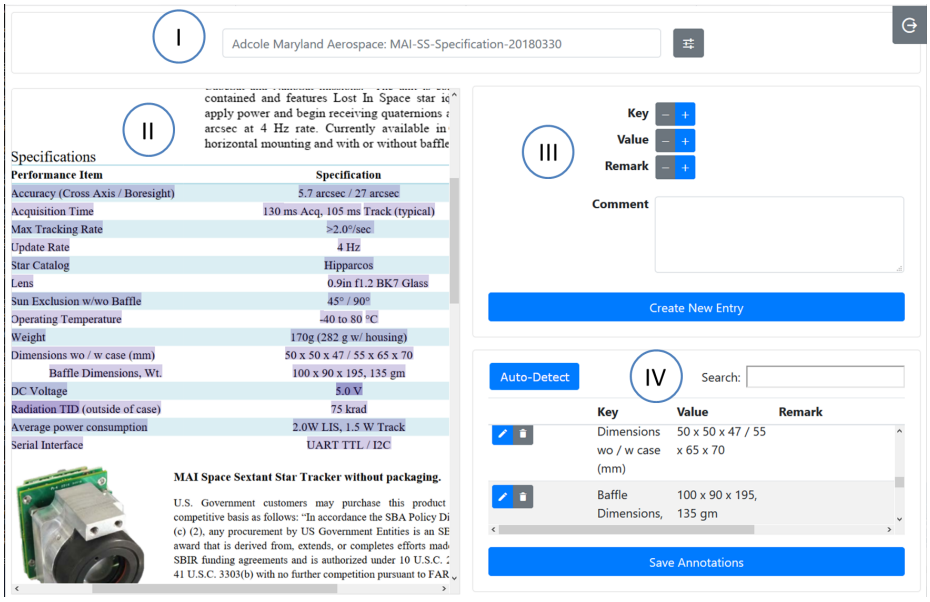

Fig. 2. Data Sheets Annotation Tool interface. It provides an interface for engineers to annotate data sheets either manually or automatically.

Fig. 2, it allows users to upload and select their data sheets (area I). The selected data sheet is displayed and attributes can be highlighted on the left panel (area II). The highlighted text can be categorized as a key, value, or remark. Engineers can add/edit/delete key-value pairs in this control form (area 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.

Manual highlighting is needed when there is no initial knowledge of data sheet's domain. If a domain-specific ontology is available, DSAT calls upon a server-side information extraction pipeline to automatically detect key-value pairs in data sheets, and the results will be highlighted on the area II. However, these detected values should be reviewed and edited by domain experts before being used further. All annotations created, both manually and automatically, are summarized on the bottom right panel (area IV).

DSAT was evaluated by targeted users, who are involved in satellite development process. The evaluation result on DSAT was collected qualitatively. The overall feedback is that DSAT is intuitive to use. However, the evaluation also revealed some areas for further improvement like supporting a wider range of data sheet formats or the ability to assign a single area of text to multiple annotations.

The manual annotations collected during this evaluation were also used to evaluate the automatic extraction by OBIE. The envisioned workflow will be presented to a wider audience to collect additional feedback and adapt it to its users' needs.

To harmonize across the heterogeneity of terms we want to search in the data sheets, we use an Ontology-Based Information Extraction (OBIE) approach. The accuracy of the auto-extracted result depends highly on the domain specific ontology. For example, when processing a star sensor data sheet, an ontology describing the specific attributes of a star sensor has to be provided. Our initial ontologies can be found in [2]. Nevertheless, the terminology used by different manufactures varies widely. Creating a comprehensive list of terms used from scratch is an almost insurmountable task given the constant evolving of the field.

To cope with this semantic challenge, we include external knowledge bases such as Wikidata [9] and WordNet [4] to expand our initial ontologies and disambiguate occurring terms[2].

We use Wikidata to find entities corresponding to occurring terms. The entities returned from Wikidata contain semantic information, e.g. alternative label, description, superclass, similar entities, etc. This information will be inserted into the initial ontology, so that it can be used to extend the scope when searching for attributes in the data sheets.

In case of multiple entities, referring to different concepts, are returned from Wikidata, we use the context of data sheets and definitions from the lexical database WordNet to disambiguate between them. WordNet encapsulates the different meanings of a word in one *synset* each. These *synsets* are composed of definition, examples of usage, and relations to other *synsets* like synonyms or hyponyms. First, we collect all domain-representing keywords from data sheets and find the *synsets* that correspond to the domain. After collecting a set of domain-representing keywords from the corpus of datasheets, the corresponding *synsets* from WordNet are retrieved. If there are multiple synsets for some keywords, the most coherent subset with respect to a semantic relatedness measure is chosen footnote In our experiments we used Wu-Palmer Similarity, but others will be explored in the future. . These synsets now provide enough context to disambiguate between the candidate entities of Wikidata by comparing the respective textual descriptions. The selected entities are then selected for enriching the ontologies.

3 Conclusions and Future Work

In this demonstration we presented DSAT, a tool to support engineers in extracting technical information from PDF data sheets. However, similar challenges as in the space domain also arise for example in patent analysis [1] or medicine [8]. Therefore, we plan to adapt DSAT to these domains. This requires a change in the underlying ontologies and possibly adaptations of the system for the structure of documents typical for those domains.

Currently, manual corrections by users only pertain to the specific data sheet they were made in. We plan to leverage this expert knowledge to further enhance

[2] The extended ontologies used in this demonstration can be found here: https://zenodo.org/record/4034478/files/enriched_ontology.zip

the ontology over time. While synonyms of existing concepts are easy to include, the situation gets more complex with hypernyms, hyponyms, or even terms that are unrelated to already known attributes. Especially for these cases we want to provide users a direct access to the ontology. Since most users have little experience with ontologies, the respective interface needs to be intuitive and prevent the users from introducing inconsistencies into the underlying ontologies.

References

1. Andersson, L., Hidir, A., Piroi, F., Allan, H.: Proceedings of The 1st Workshop on Patent Text Mining and Semantic Technologies (PatentSemTech 2019) (2019). https://doi.org/10.34726/PST2019
2. ConTrOn: Contron - spacecraft parts ontology 1.2 (May 2020). https://doi.org/10.5281/zenodo.3862854
3. (DLR), G.A.C.: Virtual satellite. `https://github.com/virtualsatellite`, accessed: 2020-08-14
4. Fellbaum, C.: WordNet : an electronic lexical database. MIT Press, Cambridge, Mass (1998)
5. Murdaca, F., Berquand, A., Kumar, K., Riccardi, A., Soares, T., Gerené, S., Brauer, N.: Knowledge-based information extraction from datasheets of space parts. In: 8th International Systems & Concurrent Engineering for Space Applications Conference (September 2018)
6. Opasjumruskit, K.: Data Sheets Annotation Tool. `https://gitlab.com/kobkaew/dsat-client`, accessed: 2020-08-17
7. Rizvi, S.T.R., Mercier, D., Agne, S., Erkel, S., Dengel, A., Ahmed, S.: Ontology-based information extraction from technical documents. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications (2018). https://doi.org/10.5220/0006596604930500
8. Starlinger, J., Kittner, M., Blankenstein, O., Leser, U.: How to improve information extraction from German medical records. it - Information Technology **59**(4) (1 2017). https://doi.org/10.1515/itit-2016-0027
9. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (Sep 2014). https://doi.org/10.1145/2629489
10. Wimalasuriya, D.C., Dou, D.: Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science **36**, 306–323 (2010). https://doi.org/10.1177/0165551509360123