INTEGRATING THE MECHANICAL DOMAIN INTO SEED APPROACH

Dworschak, Fabian; Kügler, Patricia; Schleich, Benjamin; Wartzack, Sandro

Friedrich-Alexander-Universität Erlangen-Nürnberg

ABSTRACT

Data-driven technologies have found their way into all areas of engineering. In product development they can accelerate the customization to individualized requirements. Therefore, they need a database that exceeds common product data management systems. The creation of this database proves to be challenging because in addition to explicit standards and regulations the product design contains implicit knowledge of product developers. Hence, this paper presents an approach for the semantic integration of the engineering design (SeED). The goal is an automated design of an ontology, which represents the product design in detail.

SeED fulfils two tasks. First, the ontology provides a machine-processable representation of the products design, which enables all kind of data-driven technologies. Among other representations, the ontology contains formal logics and semantics. Accordingly, it is a more comprehensible solution for product developers and knowledge engineers. Second, the detailed representation enables discovering of intrinsic knowledge, e.g. design patterns in product generations. Consequently, SeED is a novel approach for efficient semantic integration of the product design.

Keywords: Ontologies, Semantic data processing, Knowledge management

Contact:
Dworschak, Fabian
Friedrich-Alexander-Universität Erlangen-Nürnberg
Engineering Design
Germany
dworschak@mfk.fau.de

Cite this article: Dworschak, F., Kügler, P., Schleich, B., Wartzack, S. (2019) 'Integrating the Mechanical Domain into Seed Approach', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.265

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

1 INTRODUCTION

The digital transformation in product development describes the application and use of technologies from IT and communication branch within the product development process (PDP) (Matt *et al.*, 2015). The foundation of these technologies is the data, which is used to create value in terms of visualizing relationships, revealing patterns, or inferring conclusions. These data-driven technologies are particularly suitable when relationships cannot be represented by conventional, logical solutions; in other words, when a lot of intrinsic knowledge is involved. One premier example is the design in terms of the products shape. In this context, this paper presents the SeED (Semantic integration for Engineering Design) approach. The goal of the SeED approach is to provide a graph-based database, which enables to gather intrinsic knowledge about the product design. To reach this goal, the approach disassembles a mechatronic module into its individual parts, which are then categorised in domains (e.g. mechanics or electronics) and segmented until the contained information is represented as semantic subject-predicate-object statements. These triples are finally converted into an ontology. The novelty of the SeED approach lies on the one hand in the utilization of product design data for the application of data-driven technologies in engineering design and on the other hand in the detailed representation of this data. Furthermore, the resulting graph-based database enables a variety of possibilities, such as the mapping of different knowledge domains and the consistency checking of the data by formal reasoning.

The paper is structured as follows. In the next section, the problem statement and goal are carved out. After this, the state of the art with regard to applications, methods, and the semantic integration for data-driven engineering design is discussed. Thereafter, the novel method for the semantic integration in engineering design is thoroughly described, applied to a case study and then critically discussed. Finally, a conclusion and an outlook are given.

2 PROBLEM STATEMENT AND GOAL

The rapid adaptation of an existing virtual product to new customer requirements is becoming increasingly important. For concentrating on the decisive work steps, the product designer must be relieved from the routine tasks. Consequently, the use of automated processes in all phases of the PDP becomes more significant. The design of a product is an ideal application for data-driven product development. That is mainly because of two reasons. On the one hand, design patterns are repeated within the product generations or variants. The product developer can recognize these patterns by visualizing the product models. They result from standards and guidelines of the company as well as education and experience of the developer. Therefore, they are partly based on intrinsic knowledge. This leads to the second reason. As it is not possible to describe all relationships using logics and analytic functions, they can only be revealed through statistical technologies and data mining. In order for data-driven technologies to be used to customize product design, a method is required to transfer all data into an efficient and query-able (machine-processable) format. As discussed in the next section, there is a vacancy of such a format. Accordingly, the following paper presents the SeED (Semantic integration for Engineering Design) approach, which is illustrated in Figure 1.

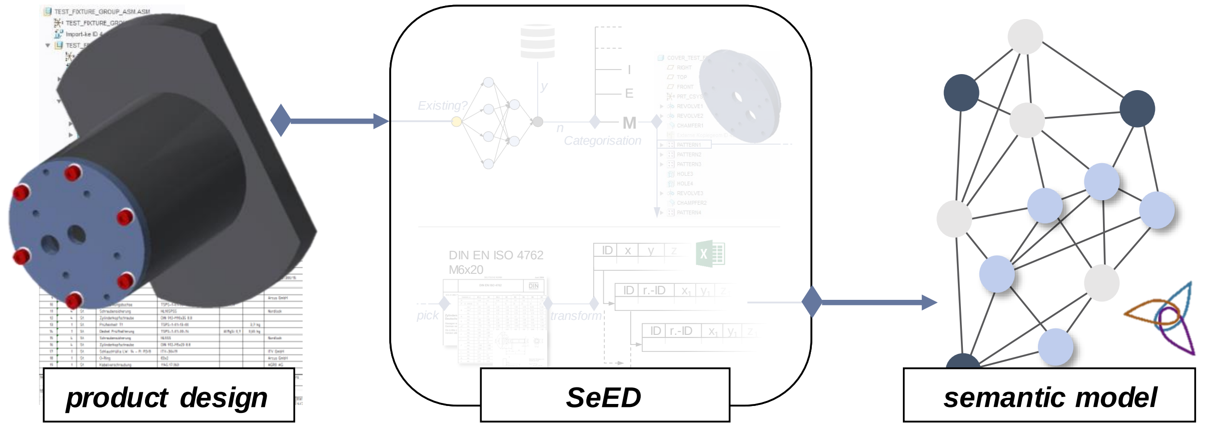


Figure 1: Initial input and result of the SeED approach

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

In order to provide an efficient and feasible solution, the SeED approach has to fulfil following challenges. First, a detailed representation of a product design contains a very large amount of data, thus SeED must be highly automated. Moreover, most products are interdisciplinary. Therefore, the approach has to provide the opportunity to link or map different semantic information from different knowledge domains. In addition to that, products change dynamically and rapidly. According to that, it must be possible to extend the representation with semantic information later on. Consequently, the goal of this paper is to show a novel approach to extract all data from the product design automatically into an ontology.

3 DATA-DRIVEN ENGINEERING DESIGN

3.1 Applications for data-driven engineering design

In the mechanical engineering industry, the use of data from all phases of the product life cycle has extensively increased in recent years. Possible areas of application include process control, predictive maintenance, customer relationship management, decision support systems and engineering design. The review paper by Harding *et al.* (2006) gives an overview of the applications at this time. It divides the applications into engineering design (10 applications presented) and manufacturing systems (28 applications presented). It is striking that significantly less examples of application were found in the field of engineering design. In addition, almost all applications in engineering design use data from other phases of the PLM to provide decision support or aid for the design phase. Only the paper of Ishino and Jin (2001) generates data from the design phase itself. There, data is collected by observing the product developers in the CAD environment. Afterwards the data is used to extract the knowledge by a method called Extended Dynamic Programming. A more recent compilation by Parraguez and Maier (2018), which presents the open data sources for engineering design used for all articles in the seven journals of the Design Society website and for all indexed papers for DESIGN and ICED, also confirms the lag of machine-processable representations of the product design. Mentioning all types of applications, only three take the design or the product structure as features for their data mining application. First, the assistance system ALARM for the development of noise-reduced machines uses the product structure and the built-in components as features to predict the acoustical properties of large rotating machines (Küstner and Wartzack, 2015). Hence, the approach focuses on the combination of components it does not include the design itself. Second, the approach by Le and Panchal (2012) models the interdependence between product and community structures by using a network-based approach. In the network describes the product structures as interfacing modules. Therefore, the approach provides information about the boundaries of each element of the product structure, but no data about the design of the individual modules. Third, the integrated Product engineering Model (iPeM) models the design process as a meta-model including the product as a system of objects (Albers *et al.*, 2010). It describes the relevant objectives, their independencies and boundary conditions. Focusing on the process, the presented meta-model does not include the design in detail. After all, the preceding list of applications for data-driven engineering design does not claim to be complete, as many applications have not been published due to privacy requirements and industrial constraints. However, the low number of applications that use the design as a data source suggests that a solution for creating machine-processable representations of the product design is still missing.

3.2 Models for data-driven engineering design

The analysis of data requires a systematic approach. The models describing a generic approach are called models for knowledge discovery in databases (KDD). They split the process into (data) pre-processing, processing and post-processing. The pre-processing addresses all step, which are applied to make the data suitable for the processing, e.g. aggregation, sampling, etc. (Tan *et al.*, 2010). The result of the pre-processing is a flat table including all the relevant data. In the processing, meta-models use the flat table as input for characterization and description of the data, association, classification, prediction, clustering, or evolution analysis. The post-processing interprets the output of the meta-models. One common implementation of these generic models is the CRISP-DM. According to the CRISP-DM, the representation of the product design is part of the business understanding and the data understanding. The first one specifies the project goals and requirements (Otte *et al.*, 2004). The second one selects and collects the required data. In the case of learning intrinsic knowledge, this

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

becomes one of the most difficult challenges to overcome due to the fact, that the design process is a multidisciplinary decision process, which includes intrinsic knowledge (Feldhusen and Grote, 2013). Therefore, a first step to use data-driven technologies for engineering design is to enable the capturing of as much details of a product as possible.

3.3 Semantic models in engineering design

For handling heterogeneous data formats and data silos, semantic technologies offer powerful tools. They provide knowledge representation and data integration, which are machine-processable on the one hand and human-understandable on the other hand. Semantic graphs were developed regarding to the work of Quillian (1967) and aim the reasonable representation of concepts and their relations (Baader, 2005). Another basic form are frame systems (Minsky, 1988), which rely on the notion of objects with attributes and on the expression of relationships between these objects (Baader, 2005). Even though these technologies are very intuitive due to their human-centred way of visualization as networks, they are not satisfactory for logical reasoning, because they provide no unique semantic characterization (Baader, 2005). Thus, ontologies can fill this lack, because they are an explicit specification of a (shared) conceptualization (Gruber, 1993) and thus they provide a representative vocabulary to describe a specific domain. In practice, the formal language for ontologies is the Web Ontology Language (OWL), which is a Description Logic (DL). DLs are usually divided in a TBox, which provides the terminological knowledge (ontology), and an ABox, which contains the assertions. The TBox is structured in a class-hierarchy with relations. The ABox contains individuals, which are assigned to the general classes from the TBox. As OWL is based on the RDF, the underlying data-structure are triples, where data entities are composed in the subject-predicate-object-scheme (Hitzler *et al.*, 2008). However, it is not only the semantic representation of knowledge, that make ontologies powerful, but rather the possibility of inference and reasoning. These mechanisms within an ontological knowledge base (KB) provide and consistency check. Moreover, inference engines ensure deduction derived from the facts, which are explicitly stated within the KB. Besides inference and reasoning, ontologies and RDF enable mapping, which means finding matches or correspondences between concepts in different data structures (Staab and Studer, 2009). In the context of mechatronic products, ALLIGATOR is an approach for mapping mechanical and electronic viewpoints of the same product (Grangel-Gonzalez *et al.*, 2016). Therefore, the approach uses an RDF scheme for identifying conflicts in between AutomationML documents, which are an Industry 4.0 standard for exchanging plant engineering information. Besides managing conflicts in exchange documents, the interoperability is a main aim for implementation of ontologies. An approach introduced by van Ruijven (2013) shows an ontology for Systems Engineering (SE). While this approach is an interpretation of SE processes (ISO 15288), it focuses on a representation of abstract processes, which improves integrity (van Ruijven, 2013). While the focus of ontologies is often on exchange and consistent representation, the analysing and synthesizing options, which ontologies provide through inference and query-accessing, is often not observed. Thus, Kestel *et al.* (2017) introduce an approach of Intentional Forgetting (IF) of ontology-based product knowledge focusing the support of the product development process. This provides relevant situational information and knowledge to the product designer through forgetting the not relevant elements in ontological KBs (Kestel *et al.*, 2017). Related to that Kügler *et al.* present a semi-automatically approach for integrating CAD-data in an ontology-based representation using Information Extraction, to provide the data and information in an ontological way for using IF. This approach aims to capture the interdependencies of assemblies and parts, but not geometric details and their relations, which describe the form and shape of the product. Thus, a finer granulated semantic model of the product is needed as a basis for applying data-driven technologies in product design.

4 SEMANTIC INTEGRATION OF THE ENGINEERING DESIGN (SEED)

As demonstrated in chapter 3.1, a machine-processable representation of the product design is still missing. Therefore, this chapter illustrates the SeED approach of how to convert a product design into a graph database (see Figure 2). The starting point is the assembly of the product. First, the assembly is disassembled into components and then the components are segmented into features and their positions (see chapter 4.1). Second, conventions are defined for the data, to serve as input for the graph database (see chapter 4.2). Third, the ontology is designed (see chapter 4.3).

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

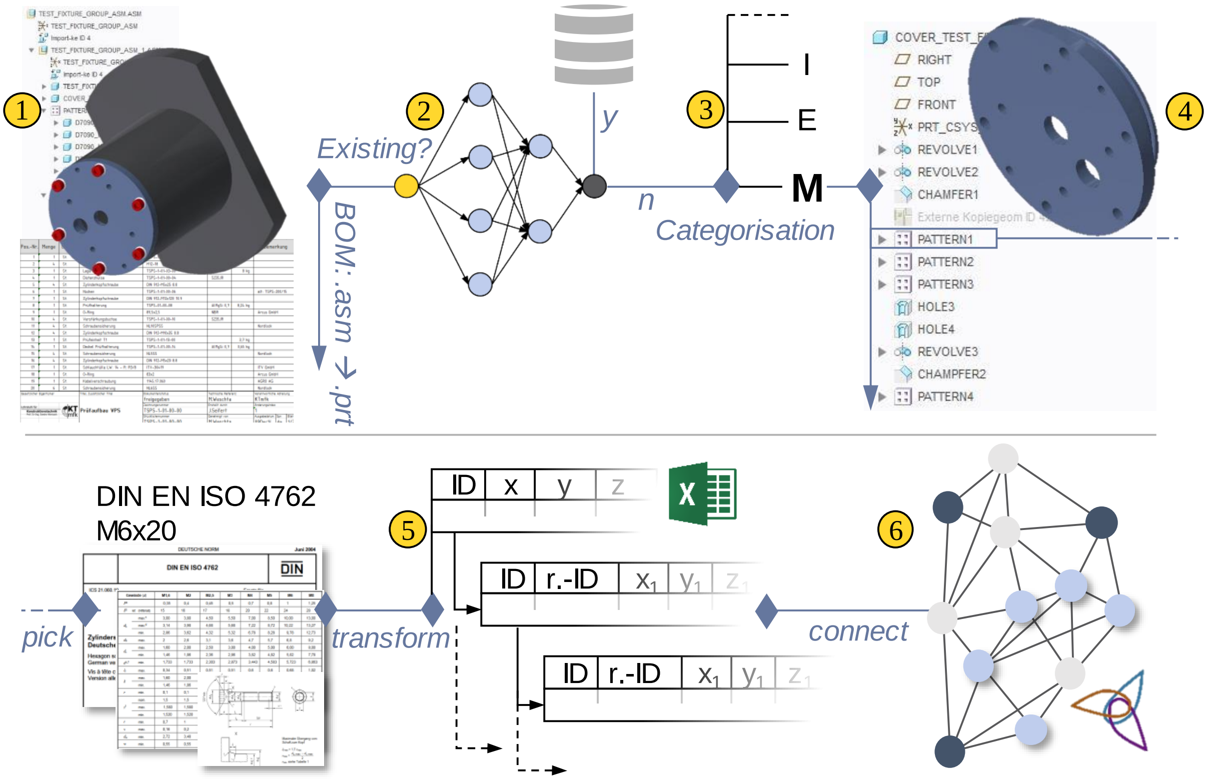


Figure 2: Overview and steps of the SeED approach

4.1 Top-down approach for disassembly and segmentation of the CAx model

Step 1: Mechatronic systems consist of assemblies or parts that can belong to multiple domains (mechanics, electronics, computer science, etc.). In order to capture the interpretation of the design in detail, the system is broken down into its components. For this purpose, a loop over the bill of material splits the system into its components until only individual parts (.prt-files) remain (see Step 1 in Figure 2). Here, a Python script iterates over the BOM. Note, that the information from the assembly's relations are lost in this procedure. They are added afterward using the approach to Kügler *et al.* (2018).

Step 2: The next loop runs over the parts and calls the part-recognition function. It detects whether the part already exists in the database (see step 2 in Figure 2). Here the method according to Spruegel and Wartzack (2015) is used. As with a FE simulation, the geometry is meshed with a fixed element type and element size. Using a principal component analysis (PCA) to rotate and shift the part into a standardized, Cartesian coordinate system. Here, the part's nodes are projected onto a detector surface with a defined size, e.g. 128x128. The number of projected nodes on each pixel of the detector surface are summed up and saved in a matrix. The matrix can be visualized as an image. Every pixel of the image is depending on the node density. These matrices train an artificial neural network (ANN) using the MATLAB® Neural Network Toolbox™. As the ANN can differ between parts, it is also possible to label parts as "known" or "unknown". As "known"-parts already exist in the database, only "unknown"-parts get a new ID and are passed on to the next step.

Step 3: Components that do not exist in the database are first categorised to domains. This step is necessary because only a domain knowledge engineer can capture and render a detailed description of the product design. The individual domains are later mapped together using the component ID in the ontology to produce the corresponding interdisciplinary links. As this process will be added to the approach later, only the mechanics domain will be illustrated in the following steps. The result of the first three steps is that a part has a unique ID and a classification into the different domains as meta-data (see step 3 in Figure 2).

Step 4: In the next step, the parts are broken down into design features. This is a hierarchical process in the model tree (see step 4 in Figure 2). Starting with the first design element at the top, the script runs over all design features. As is customary in most parametric CAD systems, a distinction is made between shapes such as: extrude, revolve, sweep, etc., and engineering features such as hole, round, chamfer, etc. The former describes the shape geometry parametrically.

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

A shape feature is defined by its type of feature, its sketch and a float number for the extrusion length or rotation angle. The sketch again consists of lines between two points, which are described by vectors and related points. In this early state of the project only extrude and sweep shapes are implemented for the ontology but other features can be added easily. In addition to the data about the feature, it is necessary to define its location. For this, the parent-child relationships of the CAD environment are used as reference elements (Kügler *et al*, 2018). The result is a one-dimensional (1D) array with all the necessary data to describe the shape feature. The entry for the sketch is a reference to another 1D-array. Chapter 4.2 illustrates this result in detail. For engineering features, it is necessary to add additional parameters. Each one has a specific a template for its 1D-array, which refers to the underlying macro.

Step 5: According to the snowflake schema, the next step of the approach stores each 1D-array in a separate file, which are named by their ID (Kemper *et al.*, 2006). In this case, .xlsx-files are used.

Step 6: The .xlsx-files provide the individuals, which are loaded in the ontology. Therefore, the Protégé-Plugin Cellfie[1] is used to automate this step. Cellfie works with transformation rules, which are written in Manchester Syntax (Horridge and Patel-Schneider, 2012) The structure of the ontology is introduced in Section 4.3.

4.2 Specification of the input for the graph-based data-base

Conventions have to be made in order for data from the design to serve as inputs to the ontology. As already mentioned, the snowflake schema defines their structure.

Shape features: As mentioned in chapter 4.1, the shape features consist of an ID, a reference surface (r-ID), the position to the reference, the type of feature, a float number for the length of the geometry and a reference to the corresponding sketch. The ID consists on the one hand of the BT-ID and on the other hand of a unique number, which was assigned by the CREO. The r-ID is read from the parent-child list, a html-file in the working directory. The next three array entries describe the position to the r-ID. The type of feature indicates whether it is an extrusion or a rotation. The next entry indicates the length of an extrusion or the angle of a rotation. The last array entry references the sketch of the shape feature. The data is stored 1D-arrays, which are illustrated in Figure 3.

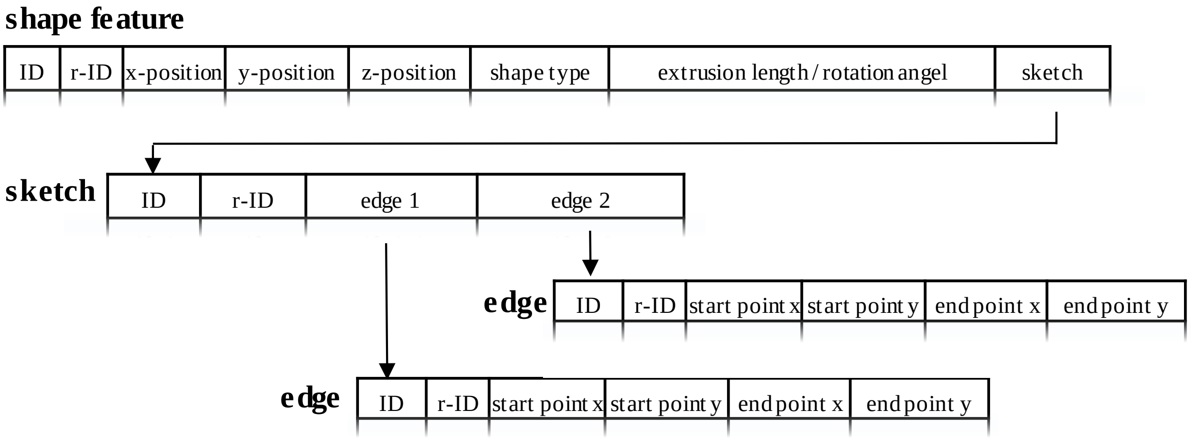


Figure 3: Overview of the extracted data from a shape feature

Sketches: Sketches are defined by their edges. Like for the shape feature the data is stored in a 1D-array. It contains the ID of the sketch, Reference ID of the shape feature and a list of the references to the edges (see Figure 3).

Edge: An edge contains of its ID, r-ID, the start-point and the end-point, described both by two entries, x, y in the local coordinate system of the reference surface of the shape. The latter is read in the shape features parent-child-list (see Figure 3).

Engineering feature: Engineering feature are described quite similar to shape features. Despite the entry for extrusion length or rotation angel. The last entry does not refer to a sketch. It refers to the engineering feature element data, e.g. a hole. The feature element data contains all the necessary information. This is illustrated in Figure 4.

[1] https://github.com/protegeproject/cellfie-plugin

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

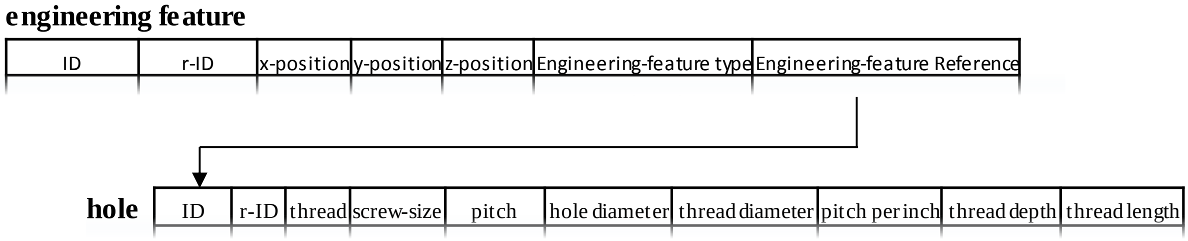


Figure 4: Overview of the extracted data from an engineering feature

4.3 Setup of the graph-based data-base

For implementing a semantic model, which is robust and extendable, it is common to use Ontology Design Patterns (ODPs). These are basic building blocks, which can be seen as reusable "Best-Practice" design solutions for different use cases and address specific Competency Questions (CQ). The basis for SeEDs ontology is the ODP SimpleORAggregated[2], which is a content ODP for the CQs "What elements are aggregated members of an object?" and vise versa. Thus, this ODP is adapted to model assembly-part-relationships (see Figure 5). Hereinafter, the focus is on the part. With the aim of modelling physical information of CAD parts, the ontology contains the specifications shape, engineering features, sketch and edge (see Section 4.2) as classes. Due to the basic data, which follows the snowflake data model (see Figure 4), a re-engineering ODP[3] is used, which provides a pattern for implement an ontology for such data structures. Following the pattern for modelling snowflake data, most of the relations between the different dimensions (lists) are defined as a "hasOrigin" object property (Figure 5, blue boxes), which is set transitive. Therefore, the CQ "Which entities are the origins of another entity?" can be answered.

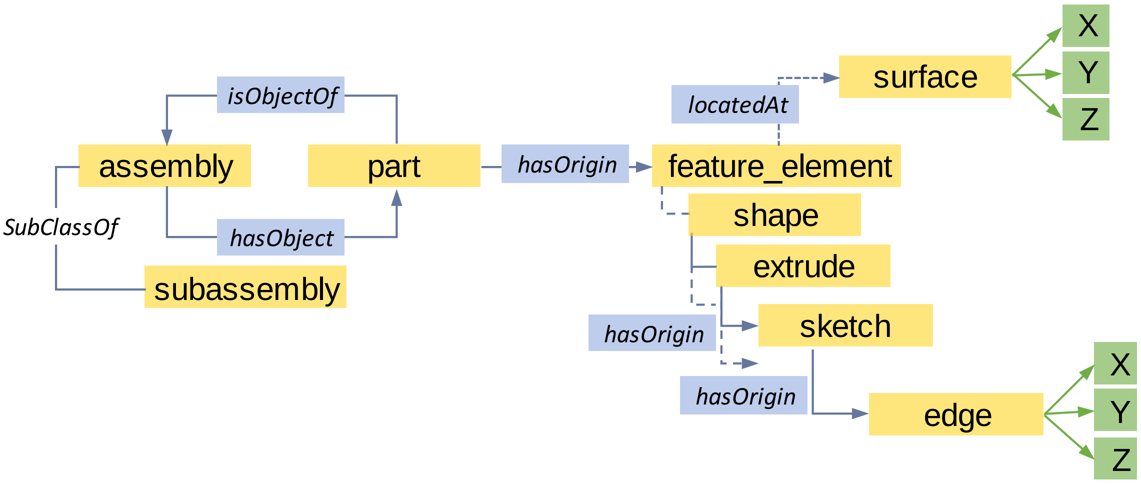


Figure 5. Schematic visualization of a section from the SeED ontology

Due to the transitivity of this relation, the part has its origin within an edge in the end. The classes "surface" and "edge" can be further located through the internal coordinates of the CAD software, which are modelled as data properties and expect a float value from the data set. Thus, the structure of the SeED ontology provides a consistent semantic model to represent and locate a part and its CAD origin.

5 EXAMPLE STUDY

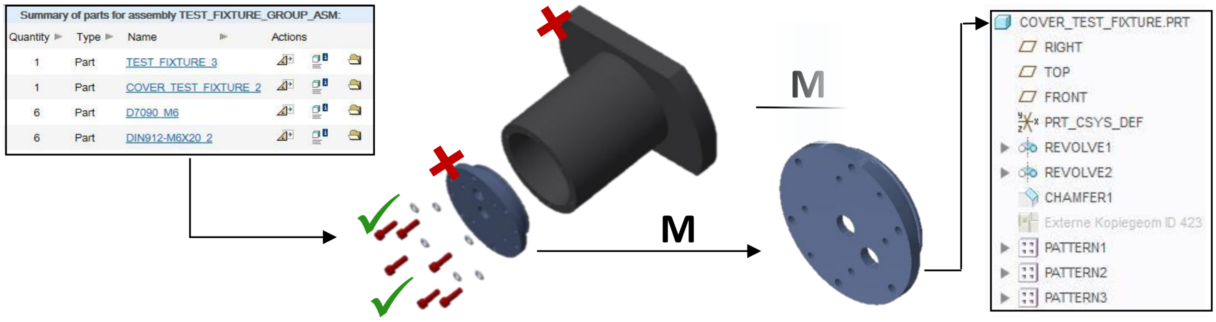
For the demonstration of the approach, a small assembly is used. It contains a cover, which is attached to the housing by six screws and their washers. In this case, the presented approach is implemented for Creo® parametric 3.0. The scripts are realized using Python (v3.7.1) and MATLAB® (R2018b). The ontology is designed in Protégé (v5.2.0). However, the approach can be realized in any other systems.

[2] http://ontologydesignpatterns.org/wiki/Submissions:SimpleOrAggregated

[3] http://ontologydesignpatterns.org/wiki/Submissions:Pattern_for_re-engineering_a_classification_scheme%2C_which_follows_the_snowflake_data_model%2C_into_an_ontology_schema

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

In Figure 6, the assembly is split in its individual parts, using the bill of materials. Then the part recognition tool labels the parts in “know” parts symbolled by green ticks and “unknown” parts symbolled by red crosses. As there are only parts of the mechanical domain, the two unknown parts (cover and housing) are categorised to this knowledge domain. For the following, the cover is used to show the rest of the approach. As the following section will show, the cover itself has 3 shape features and 29 engineering features, which lead to a total number of 89 tables as input for the ontology. An easily manageable effort for one part.

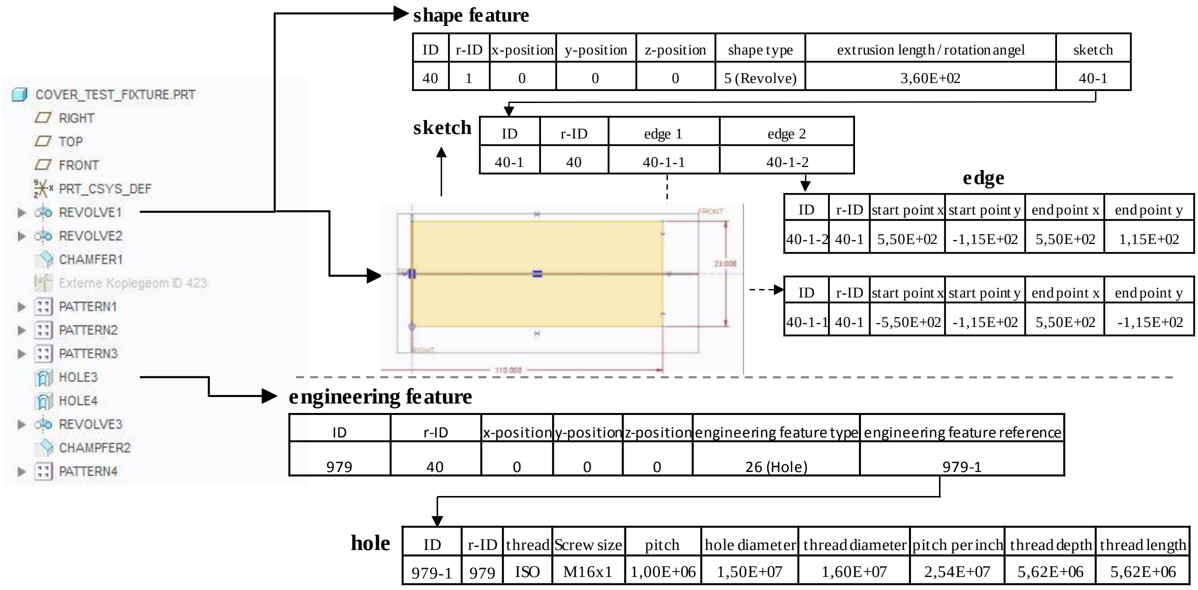


Summary of parts for assembly TEST_FIXTURE_GROUP_ASM:

Quantity	Type	Name	Actions
1	Part	TEST_FIXTURE_3	
1	Part	COVER_TEST_FIXTURE_2	
6	Part	D7090_M6	
6	Part	DIN912-M6X20_2	

Figure 6: Steps 1 - 3 of the SEED approach (Spitting the BOM, labelling of known parts and categorisation to knowledge domains)

On the part level, the Python script runs over the model tree, which is stored as an .html-file in the working directory of the part. As shown in Figure 7, the part contains three shape features (Revolve 1-3). According to the parent-child-relations, REVOLVE1 is referenced to the TOP plane and its position is defined by the global coordinate system (x=0.00; y=0.00; z=-15.00). Note that all units are in millimetre, Newton, second (mmNs). The rotation angel of REVOLVE1 is 360° and the sketch is referenced as 40-1. The sketch contains two edges. All data is collected by the script and saved in separate .xlsx-files.



shape feature

ID	r-ID	x-position	y-position	z-position	shape type	extrusion length / rotation angel	sketch
40	1	0	0	0	5 (Revolve)	3,60E+02	40-1

sketch

ID	r-ID	edge 1	edge 2
40-1	40	40-1-1	40-1-2

edge

ID	r-ID	start point x	start point y	end point x	end point y
40-1-2	40-1	5,50E+02	-1,15E+02	5,50E+02	1,15E+02

ID	r-ID	start point x	start point y	end point x	end point y
40-1-1	40-1	-5,50E+02	-1,15E+02	5,50E+02	-1,15E+02

engineering feature

ID	r-ID	x-position	y-position	z-position	engineering feature type	engineering feature reference
979	40	0	0	0	26 (Hole)	979-1

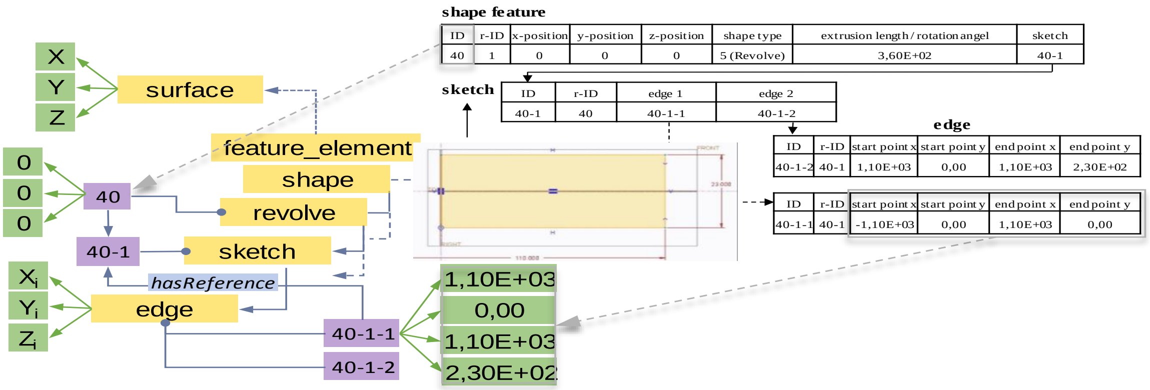
hole

ID	r-ID	thread	Screw size	pitch	hole diameter	thread diameter	pitch per inch	thread depth	thread length
979-1	979	ISO	M16x1	1,00E+06	1,50E+07	1,60E+07	2,54E+07	5,62E+06	5,62E+06

Figure 7: Steps 4 and 5 of the SeED approach (Extraction of features and saving the data)

Once the data is stored in the .xlsx-files, the Protégé-plugin Cellfie can be used to instantiate the data. For the individuals the ID-numbers are used as names and the position data can be transferred to the data properties (see Figure 8 green boxes). To store the references between the data dimensions, the relation “hasReference” is applied, e. g. the edge “40-1-1” hasReference “40-1” (see Figure 8). Thus, all the references can be queried, thus all relationships are transparent. The semantic model of the data provides a query- and machine-processable representation. The logical consistency is checked by reasoning mechanisms. Through the relations and properties, inference mechanisms can be used to derive effects caused by changes within the semantic model. The resulting instances and data properties of the ontology (see purple and green boxes on the left of Figure 8) build the semantic model of the form and shape as well as the hierarchical interdependencies of the product. Thus, the approach results in a consistent machine-processable description of the cover in OWL, which builds the basis for data-driven technologies.

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press



shape feature

ID	r-ID	x-position	y-position	z-position	shape type	extrusion length / rotation angel	sketch
40	1	0	0	0	5 (Revolve)	3,60E+02	40-1

sketch

ID	r-ID	edge 1	edge 2
40-1	40	40-1-1	40-1-2

edge

ID	r-ID	start point x	start point y	end point x	end point y
40-1-2	40-1	1,10E+03	0,00	1,10E+03	2,30E+02

ID	r-ID	start point x	start point y	end point x	end point y
40-1-1	40-1	-1,10E+03	0,00	1,10E+03	0,00

Figure 8: Steps 6 of the SeED approach (Integration of the data within the semantic model)

6 DISCUSSION

As chapter 3.1 illustrated, there is a lack of data-driven applications using data from the product design. This void is induced by missing machine-processable models for the engineering design of a product. This data is urgently needed to make the product development process part of digital transformation. In order to close this gap, chapter 4 has shown an approach that maps the design of a product into an ontology. The procedure accomplishes this task automatically and can therefore be applied to more complex components, assemblies or parts with a reasonable amount of time and effort. When using the approach, the following challenges must be considered. First, the approach is based on parametric features. This requires a fully parametric description of the parts and leads to two challenges. On the one hand, an individual script must be created for each shape or engineering feature. The description of the features is also dependent on the CAD tool used. Both means considerable effort in implementing the approach in a platform-independent tool. For the presented approach, the CREOs STEP import was used. For the future, it is recommended to extract the data from the STEP files directly. Furthermore, sub-steps of the procedure cause considerable effort, for example the training of the part recognition in step 2 in Figure 2. This becomes particularly clear when it is necessary to map many different parts in the ontology for the digital transformation of the product development process. Therefore, it is recommended to use the approach, when the details of the engineering design matter. A predestined use case for the presented approach is the automation of adaptations in the product designs across generations or variations. An example is the individualization of products. For this customization of the product design, the spectrum of features does not vary, so the implementation effort is very limited. In order to automatize the adaptions, link the ontology of the products to their requirement configuration. For new requirement configurations, variants can be generated automatically with the help of data mining techniques.

7 CONCLUSION AND FUTURE WORK

While in other stages of the product lifecycle many sources of engineering data exist, there are still very few sources available in the design phase. The SeED approach presented in the previous paper shows a way to extract data from the product design itself. The approach builds around the design of an ontology, which contains detailed data of a parametric product design. Therefore, the SeED approach bridges the mentioned gaps in design data. So far, the approach is limited to one knowledge domain (mechanics). This constraint needs to be broken by building and integrating ontologies for additional domains (e.g. electronics). In addition, information about the structure of the systems is lost when the modules are divided into components (see Step 1 in Figure 2). Here, the parent-child relationships at the assembly level need to be added to the ontology. If these two extensions are integrated, the ontology is a complete description of geometrical and physical model of the product. The other levels (requirements, functions and logics) have already been digitized with the help of the modeling language SysML. This representation is based on .xml files, which are also to be integrated into the ontology.

REFERENCES

Albers, A., Ebel, B. and Sauter, C. (2010), "Combining process model and semantic wiki", 11th International Design Conference, Dubrovnik, Croatia, May 17-20, Design Society, Glasgow, UK, pp. 1275–1284.

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press

Baader, F. (2005), The description logic handbook: Theory, implementation, and applications, Cambridge Univ. Press, Cambridge. http://dx.doi.org/10.1017/CBO9780511711787.

Feldhusen, J. and Grote, K.-H. (2013), Pahl/Beitz Konstruktionslehre, Springer, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-642-29569-0.

Grangel-Gonzalez, I., Halilaj, L., Auer, S., Lohmann, S., Lange, C. and Collarana, D. (2016), "An RDF-based approach for implementing industry 4.0 components with Administration Shells", 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 06.-09.09.2016, IEEE, pp. 1–8. http://dx.doi.org/10.1109/ETFA.2016.7733503.

Gruber, T.R. (1993), "A translation approach to portable ontology specifications", *Knowledge Acquisition*, Vol. 5 No. 2, pp. 199–220. http://dx.doi.org/10.1006/knac.1993.1008.

Harding, J.A., Shahbaz, M. and Kusiak, A. (2006), "Data mining in manufacturing. A review", *Journal of Manufacturing Science and Engineering*, Vol. 128 No. 4, pp. 969–976.

Hitzler, P., Krötzsch, M., Rudolph, S. and Sure, Y. (2008), Semantic Web, Springer, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-540-33994-6.

Horridge, M. and Patel-Schneider, P.F. (2012), "OWL 2 Web Ontology Language", *Manchester Syntax* (2nd ed.). [online] W3C, Available at: https://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/ (accessed 19 February 2018).

Ishino, Y. and Jin, Y. (2001), "Data mining for knowledge acquisition in engineering design", In Data mining for design and manufacturing, Springer, Berlin, Heidelberg, pp. 145–160.

Kemper, H.-G., Mehanna, W. and Unger, C. (2006), Business Intelligence - Grundlagen und praktische Anwendungen: Eine Einführung in die IT-basierte Managementunterstützung, IT erfolgreich lernen, 2nd ed., Vieweg and Sohn Verlag, Wiesbaden. http://dx.doi.org/10.1007/978-3-8348-9056-6.

Kestel, P., Luft, T., Schon, C., Kügler, P., Bayer T., Schleich, B. and Wartzack, S. (2017), "Konzept zur zielgerichteten, ontologiebasierten Wiederverwendung von Produktmodellen". *Design for X*, 04.-05.10.2017, TuTech, Hamburg, pp. 241–252.

Kügler, P., Kestel, P., Schon, C., Marian, M., Schleich, B., Staab, S. and Wartzack, S. (2018) "Ontology-based approach for the use of the intentional forgetting in product deveopment", In Proceedings of the DESIGN, pp. 1595–1606. http://dx.doi.org/10.21278/idc.2018.0402.

Küstner, C. and Wartzack, S. (2015), "The realization of an engineering assistance system for the development of noise-reduced rotating machines". 20th International Conference on Engineering Design, Milan, Italy, 27.-30.07.2015, Design Society, Glasgow, UK, pp. 71–80.

Le, Q. and Panchal, J.H. (2012), "Analysis of the interdependent co-evolution of product structures and community structures using dependency modelling techniques", *Journal of Engineering Design*, Vol. 23 No. 10–11, pp. 807–828.

Matt, C., Hess, T. and Benlian, A. (2015), "Digital Transformation Strategies", *Business and Information Systems Engineering*, Vol. 57 No. 5, pp. 339–343. http://dx.doi.org/10.1007/s12599-015-0401-5.

Minsky, M. (1988), "A framework for representing knowledge", In: Haugeland, J. (Ed.), *Mind design: Philosophy, psychology, artificial*, 6th ed., MIT Press, Cambridge.

Otte, R., Otte, V. and Kaiser, V. (2004), *Data Mining für die industrielle Praxis*, Hanser, München.

Parraguez, P. and Maier, A. (2018), "Data-driven engineering design research. Opportunities using open data". *21st International Conference on Engineering Design*, Vancouver, Canada, 21-25.08., Design Society, Glasgow, UK, pp. 41–50.

Quillian, M.R. (1967), "Word concepts. A theory and simulation of some basic semantic capabilities", *Behavioral science*, Vol. 12 No. 5, pp. 410–430. http://dx.doi.org/10.1002/bs.3830120511.

Spruegel, T.C. and Wartzack, S. (2015), "Concept and application of automatic part-recognition with artificial neural networks for FE simulations". *20th International Conference on Engineering Design*, Milan, Italy, 27.-30.07.2015, Design Society, Glasgow, UK, pp. 183–194.

Staab, S. and Studer, R. (2009), *Handbook on Ontologies, International Handbooks on Information Systems*, 2nd ed., Springer, Berlin, Heidelberg. http://dx.doi.org/10.1007/978-3-540-92673-3.

Tan, P.-N., Steinbach, M. and Kumar, V. (2010), *Introduction to data mining, int.* ed., Pearson/Addison-Wesley, Boston, Mass.

van Ruijven, L.C. (2013), "Ontology for Systems Engineering", *Procedia Computer Science*, Vol. 16, pp. 383–392. http://dx.doi.org/10.1016/j.procs.2013.01.040.

ACKNOWLEDGEMENT

This research work is part of the "Optimization-based design methodology in early phase of mechatronic product development" project and funded by the European Regional Development Fund (ERDF/OptMePro). The authors also gratefully acknowledge the financial support of project WA 2913/22-1 within the Priority Program 1921 by the German Research Foundation (DFG)

https://doi.org/10.1017/dsi.2019.265 Published online by Cambridge University Press