[CJW+22] B. Caesar, N. Jansen, M. Weigand, M. Ramonat, C. S. Gundlach, A. Fay, B. Rumpe:
Extracting Functional Machine Knowledge from STEP Files for Digital Twins.
In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), pp 1-4. , IEEE, September 2022.
www.se-rwth.de

Extracting Functional Machine Knowledge from STEP Files for Digital Twins

Birte Caesar[1], Nico Jansen[2], Maximillian Weigand[1], Malte Ramonat[1],
Claas Steffen Gundlach[1], Alexander Fay[1], Bernhard Rumpe[2]

2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA) | 978-1-6654-9996-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ETFA52439.2022.9921451

***Abstract*—The current challenges of industrial manufacturing forces producers to optimize and to digitize their facilities. The Digital Twin as a digital representation of both the product and the production is a key enabler to efficiency, flexibility, and sustainability. Unfortunately, the development of Digital Twins is sophisticated and hampered by manual tasks. This paper presents an approach to automatically create digital models of the objects which are to be represented, based on 3D CAD data. Therefore, the CAD data, which is stored as a STEP file, is analysed to extract relevant information for the following graph analysis, which is used to identify components, their dependencies and the resulting functional modules. The graph analysis results will be used in future work to implement a digital twin.**

***Index Terms*—CAD data, Digital Twin creation, graph analysis**

I. Introduction

The manufacturing industry faces increased competition due to lower market entry barriers as a result of globalization and individualized customer demands. Further, shortened product life cycles and innovation cycles, as well as disrupted manufacturing operations and supply chains, a problem aggravated by the COVID-19 pandemic, require flexible and adaptable production systems [1]. To keep companies competitive, entire value chains need to be optimized and automated based on digitally available information. In this context, the topic of digital twins (DTs) receives increasing attention, as a study by the market research company Gartner shows that 62 percent of interviewed companies plan to implement DTs and 13 percent already use DTs [2]. To develop a DT, first its desired characteristics and purpose need to be defined. The followed implementation is usually carried out specifically for the respective application and is not integrated with the development of the system [3]. Especially in the case of brownfield systems, where the DT is created by operators independently from the system development, the DT modeling effort is increasingly high as the virtual representation of real entities must be completely remodeled [4]. Thus, engineering of DTs can be time-consuming and complicated [3].

In addition, there are plenty of brownfield systems that will be in use for several upcoming years but need to be extended by functionalities, e.g., optimized operating points and reconfiguration management. For these systems, different engineering documents are available, varying in quality and information content. In most cases, 3D CAD drawings are available. These drawings represent the hardware components of the system. These components fulfill the mechanical process, whereas software components control these processes. The mechanical process can be divided into several functions, which are fulfilled by modules. This results in the following research question: *how can modules of hardware components be automatically identified, based on common engineering artifacts, and transformed into a model with a suitable degree of formalization so that the manual modeling effort for the creation of DTs for existing and in use systems is reduced?*

Therefore, an approach to support the creation of DT models based on common engineering artifacts, in particular 3D CAD drawings, is presented in this paper. The boundary representation of STEP files that represent the minimum information content required to be able to extract mechanical processes is used for this purpose. Components, their contact faces, and the degree of freedom (DOF) that any two components that share a contact face have to each other are extracted from STEP files. Finally, knowledge about clustering the components into modules is inferred, which builds the basis for control groups.

The remainder of the paper is structured as follows. Sec. II presents an overview of related work, followed by Sec. III, presenting the concept of extracting knowledge of the hardware components based on STEP files, to transfer the knowledge into an ontology and infer the functional module. The paper concludes with a summary and outlook in Sec. IV, where the open work and the planned procedure is outlined.

II. Related Work

Our work is based on the **ST**andard for the **E**xchange of **P**roduct (STEP) model data, developed by the International Standards Organization and documented in ISO 10303. It is an exchange format designed to include all product-related information along the product life cycle, widely established in the manufacturing industry. It enables the consistent and error-free usage of 3D CAD data across different (often proprietary) tools. Amongst others, the standard describes the product in the form of an assembly structure [5] as well as a boundary representation [6]. This section covers the format and related work of transforming its data into a knowledge base.

Ontologies are used to formalize knowledge in a machine-interpretable fashion. They enable a semantic interoperability between systems and reduce the need for manual and error-prone knowledge alignment [7]. Moreover, the interpretability

[1]Institute of Automation Technology, Helmut-Schmidt-University, Hamburg, Germany, email: birte.caesar@hsu-hh.de
[2]Software Engineering Department of Computer Science 3, RWTH Aachen University, Aachen, Germany, email: jansen@se-rwth.de

of knowledge can be utilized to infer new knowledge automatically by using so-called reasoners. A prominent specification for modeling ontologies is the Web Ontology Language (OWL). It formalizes terminological knowledge about concepts, relations, and attributes in the terminological box (TBox). Similarly, assertional knowledge about individuals of the ontology is formalized in the assertional box (ABox), following the rules specified in the TBox [7].

Barbau et al. [8] created an ontology to represent the concepts of the STEP standard as well as an algorithm to automatically transform the STEP data into ABox individuals. The goal of the approach was to combine different ontology TBoxes to combine product data from different sources and thus to enable a continuous product life cycle management. Amongst others, the core product model and the open assembly model, developed by the National Institute of Standards and Technology, can also be included as a TBox to obtain information which are not provided by the STEP file data, e.g., function and behavior description. Perzylo et al. [9] follow a similar approach, but are only considering the boundary representation of objects. This enables the transformation of STEP and IGES files into the so-called OntoBrep ontology. The aim of the approach is to link further information to the product data, in particular constraints for a robot assembly task description. Gong et al. [10] extend the work of Barbau et al. [8] with a focus on extracting assembly relations from the STEP files for a better assembly sequence planning.

Other solutions deal with the semantic annotation of 3D CAD models for better reuse during engineering design. Qin et al. [11] present an approach based on component feature recognition, where each feature fulfills a function. The functions and features are formalized in an ontology TBox. Lupinetti et al. [12], [13] take the analyses even further by extracting the structure of assemblies and the possible movement of each assembly component. This allows a comparison of components based on the kinematic function, which is a common cluster for reuse. Han et al.'s [14] approach is based on Lupinetti et al. [12], transferring the same extracted information into an ontology and extending it by a more detailed functional description of the components. Vilmart et al. [15] present a similar approach focusing on cellular models, where single components are merged and information about contact faces is missing.

A third approach is based on extracting structural, kinematic, or functional information of product assemblies for further usage in, e.g., virtual commissioning. Thongnuch et al. [16] present a method for the automatic generation of kinematic simulation models from mechanical CAD assembly models for virtual commissioning of mechatronic systems. Components of the assembly model and assembly constraints, i.e., kinematic relations between the components, are extracted and mapped to kinematic joints. Based on the joints and constraints, motion vectors of the machine components are derived and combined based on a fixed root component, resulting in a range of kinematic motion of the machine. This generated range of motion and geometrical information is formulated as COLLADA model for standardized geometry and kinematics information exchange. Further, Hildebrandt et al. [17] present a method to extract kinematic skills, i.e., abilities to implement a production-related process, from 3D-CAD assembly data of a robotic gripping unit to assist engineers in checking its functionality. The method utilizes movement restrictions and materials to infer kinematics, reachable positions and a maximum payload. The information is mapped into a target ontology by means of a rule set. The ontology assigns reachable positions to assembly parts and combines movement vectors of parts to movement descriptions of the system. Kinematic skills are inferred from these descriptions in combination with production system information.

The related work shows potential use cases for extracting information from 3D CAD models. However, the presented approaches cannot be applied to all STEP models since some assembly information contained therein and the DOF of components is not exported into STEP files from proprietary CAD tools. In general, the limitations apply to the usage of proprietary CAD tools. Therefore, using a vendor-neutral data format such as STEP as a data basis is desirable, but the presented methods do not support STEP data or rely on proprietary CAD tools.

III. Methodology to extract Functional Knowledge

In this section the methodological procedure to identify the atomic functions of the examined systems as well as the classification of the system components into modules is described. To create a uniform understanding, atomic functions are defined as the ability of two components to enable a relative movement to one another. This definition can be considered analogous to the definition of capabilities in [18]. Relative movement can be enabled in rotational or translational direction. A combination of enabling rotational or translational movement between two components would result in two atomic functions. A complex movement, involving multiple components, can therefore be divided into multiple atomic functions. The procedure can be clustered into three main tasks. First, extracting the relevant data from the STEP file and analysing the contact faces of the components regarding their DOF. Second, transforming the gathered information into an ontology in order to perform the third task of graph analysis to identify the modules.

A. Extracting Components and Degree of Freedom

Generally, STEP provides a mere topological and geometrical export of the product. The topology describes adjacency relationships between objects and the position and arrangement of geometric objects, while the geometry is the mathematical description of the object itself. However, knowledge of the functional hierarchy is hidden within the assembly structure and not explicitly provided. To extract functional relationships of components, the possible interactions between the different solids concerning their mutually relative position and shape need to be analyzed. Therefore, the geometry of atomic elements of the solids, i.e., their faces, edges, and vertices, are analyzed for translational and rotational constraints.

Translational restrictions result from at least two faces of different components being in contact. The translational direction in which they interlock is given by the normal vector of the corresponding faces. For a rotational connection, cylindrical surfaces around a shared axis and with the same radius must mesh to form a joint. This results in a rotational constraint with a corresponding rotation axis. Another possibility emerges from spherical constructs, which do not yield a single fixed rotation axis. For components without interlocking faces, no constraints are derived. After the constraints between two respective components have been identified, the DOF of each component within the structure must be determined. The DOF is defined by the number of independent variables of a system, which mechanically characterize the independent motions of a body. For each component of the structure under investigation, the DOF is expressed by its translational and rotational possibilities of motion. The translations are described by the respective motion vector. Rotation is determined by specifying the axis of rotation by means of the location and direction vector. The reference of all location and direction specifications is the origin of the global coordinate system, which is given by the examined STEP file [19].

To determine the DOF of every component within a structure, all constraints of it must be analyzed and summed up. Initially, each component has six DOFs: translational motions along and arbitrary rotations about the three global coordinate axes. These DOFs are now limited by the previously determined constraints through the connections to other components within the structure. The constraints can thereby have an impact on the translational and rotational freedoms and are transferred accordingly. After evaluation of all constraints of a component one receives its remaining degree of freedom.

B. Ontology Transformation

Once the components and their DOFs have been extracted and computed from the STEP file, the information is transformed into a graph. As mentioned in Sec. II, graphs and ontologies are highly suitable to represent engineering artifacts, especially if new knowledge has to be inferred (cf. Sec. III-C).

Fig. 1 shows the TBox the graph analysis is based on. It is based on an ontology design pattern (ODP) representing the guideline VDI 2206. The general idea of creating reusable, modular TBoxes and adapt and extend them according to project specific requirements, was defined in [20]. In this paper, the ODP of the VDI 2206 established in [7] is reused. To describe the DOFs of two components, the TBox has to be extended (cf. Fig. 1). While grey boxes represent the existing ODP, green boxes and green highlighted relations depict new elements.

The extension generally consists of the new class *DOF* and its two subclasses *TranslationalDOF* and *RotationalDOF*. Instances of the *MechanicalInterface* can be associated with instances of *DOF* via the new object property *hasDOF*. Each *DOF* is defined by a directional vector and, in case of a rotational DOF, a location vector. Those vectors are defined by the data properties *directionalVectorX,-Y,-Z* and *locationVectorX,-Y,-Z*, which are directly associated with the corresponding instances.

The extracted solids will be created as individual of the class *Component*. For each identified contact of two solids, a *MachanicalInterface* is instantiated. The respective components are related to the mechanical interface via the object property *hasInterface*. If the two components can move relatively to each other, a translational or rotational *DOF* instance is created. The next subsection explains how the individuals of the class *Module* are inferred. The remaining classes of the ODP are not considered in this approach.

C. Graph Analysis

The goal of the graph analysis is to group the system components into modules. Each module includes at least the components that contribute to fulfill one atomic function. As described above, an atomic function enables the movement of two components relative to each other. First, the interfaces that are related to a DOF are selected. The following two steps are executed for each interface related to a DOF separately. Second, components which are related to a selected interface are selected. Third, all associated interfaces of the components are selected. Steps two and three are repeated until an interface that is already identified to have a DOF is reached. The component related to that interface is not included in the module. The result of the selection are several modules, which could be overlapping. Components represented in more than one module are connecting components. These are components that have an interface without DOF to components that either have a connection to an interface with DOF or not. Connecting components link atomic functions and therefore create joined functions. Connecting components are excluded from the modules as they were included in at the first place.

Accordingly, an individual for the module is created in the graph and each including component is related to it with the object property *consistsOf*. The graph query language SPARQL is used in order to traverse the graph. To be capable to define joined functions, further rules need to be specified. To do so, more examples will be explored. Therefore, the goal of our future work is to present an extended rule set to define modules that provide joined functions.

IV. Summary & Outlook

This paper presented a procedure to extract relevant information to infer knowledge to cluster system components into modules which represent mechanical functions. The implementation to extract and identify the DOF as well as transformation into an ontology has been completed. Several different STEP files have been used to verify the results, but a deeper evaluation with industry examples will have to be accomplished, as several difficulties based on the general nature of STEP files have been discovered. First of all, STEP files are very heterogeneous, depending on the 3D CAD tool used by the creator of the drawing and the way how

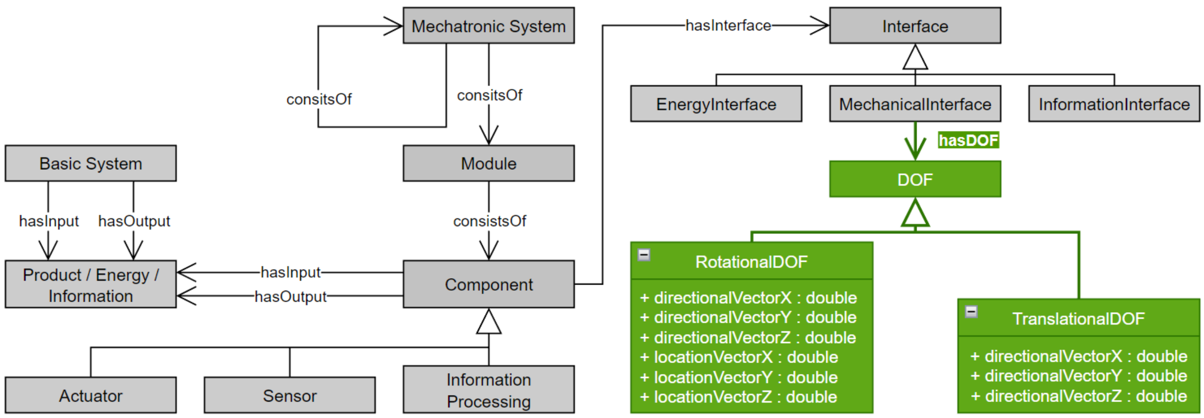

Fig. 1: Extended Ontology Design Pattern TBox of VDI 2206

the drawing is created. Therefore, a polygon's normal vector sometimes points inwards of the solid to which it belongs, ultimately resulting in not detecting two faces as being in contact. Further, typical CAD modeling involves instantiating the same component several times. In the product hierarchy, these components are represented individually, but the STEP boundary representation only contains the component description once. While the STEP standard provides unambiguous interpretation, this issue impedes the extraction of individual components. Ultimately, we strive to solve existing issues and publish the finalized proof of concept in detail.

Furthermore, the rule set to infer the structural hierarchy of the components has been described, but has to be implemented and evaluated. After finishing the implementation and evaluation, the work on the meta model to relate the structural system hierarchy with its functional units will start. Based on the meta model, it will be possible to automatically generate the functional system architecture for the involved machine components. This model can then be further extended manually with ports for sensor and actor signals if these cannot be extracted from the CAD data. Once the model is completed, the digital twin can be used as shown in previous work [3].

References

[1] J. Morgan, M. Halton, Y. Qiao, and J. G. Breslin, "Industry 4.0 smart reconfigurable manufacturing machines," *Journal of Manufacturing Systems*, vol. 59, pp. 481–506, 2021.

[2] Costello, Katie; Omale, Gloria, "Gartner survey reveals digital twins are entering mainstream use," February 2019.

[3] P. Bibow, M. Dalibor, C. Hopmann, B. Mainz, B. Rumpe, D. Schmalzing, M. Schmitz, and A. Wortmann, "Model-driven development of a digital twin for injection molding," in *Advanced information systems engineering*, vol. 12127 of *LNCS Sublibrary: SL 3, Information Systems and Applications, incl. Internet/Web, and HCI*, pp. 85–100, Springer, 2020.

[4] D. Braun, M. Riedhammer, N. Jazdi, W. Schloegl, and M. Weyrich, "A methodology for the detection of functional relations of mechatronic components and assemblies in brownfield systems," *Procedia CIRP*, vol. 107, pp. 119–124, 2022.

[5] ISO 10303-44, "Industrial automation systems and integration - product data representation and exchange - part 44: Integrated generic resource: Product structure configuration," 2019.

[6] ISO 10303-42, "Industrial automation systems and integration - product data representation and exchange - part 42: Integrated generic resource: Geometric and topological representation," 2019.

[7] C. Hildebrandt, A. Köcher, C. Küstner, C.-M. López-Enríquez, A. W. Müller, B. Caesar, C. S. Gundlach, and A. Fay, "Ontology building for cyber–physical systems: Application in the manufacturing domain," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1266–1282, 2020.

[8] R. Barbau, S. Krima, S. Rachuri, A. Narayanan, X. Fiorentini, S. Foufou, and R. D. Sriram, "Ontostep: Enriching product model data using ontologies," *Computer-Aided Design*, vol. 44, no. 6, pp. 575–590, 2012.

[9] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for cad data and geometric constraints as a link between product models and semantic robot task descriptions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015)*, (Piscataway, NJ), pp. 4197–4203, IEEE, 2015.

[10] H. Gong, L. Shi, D. Liu, J. Qian, and Z. Zhang, "Construction and implementation of extraction rules for assembly hierarchy information of a product based on ontostep," *Procedia CIRP*, vol. 97, pp. 514–519, 2021.

[11] F. Qin, S. Gao, X. Yang, M. Li, and J. Bai, "An ontology-based semantic retrieval approach for heterogeneous 3d cad models," *Advanced Engineering Informatics*, vol. 30, no. 4, pp. 751–768, 2016.

[12] K. Lupinetti, F. Giannini, M. Monti, and J.-P. Pernot, "Automatic extraction of assembly component relationships for assembly model retrieval," *Procedia CIRP*, vol. 50, pp. 472–477, 2016.

[13] K. Lupinetti, F. Giannini, M. Monti, and J. Pernot, "Cad assembly descriptors for knowledge capitalization and model retrieval," in *Tools and methods of competitive engineering*, Delft University of Technology, 2016.

[14] Z. Han, R. Mo, H. Yang, and L. Hao, "Cad assembly model retrieval based on multi-source semantics information and weighted bipartite graph," *Computers in Industry*, vol. 96, pp. 54–65, 2018.

[15] H. Vilmart, J.-C. Léon, and F. Ulliana, "From cad assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model," *Computer-Aided Design and Applications*, vol. 15, no. 3, pp. 300–317, 2018.

[16] S. Thongnuch and A. Fay, "Mcad2sim: Towards automatic kinematic joints recognition," *Computer-Aided Design and Applications*, 2020.

[17] C. Hildebrandt, M. Glawe, A. W. Müller, and A. Fay, "Reasoning on engineering knowledge: Applications and desired features," in *European Semantic Web Conference*, vol. 10250 of *Lecture notes in computer science*, pp. 65–78, Springer International Publishing, 2017.

[18] E. Järvenpää, P. Luostarinen, M. Lanz, and R. Tuokko, "Presenting capabilities of resources and resource combinations to support production system adaptation," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, pp. 1–6, IEEE, 2011.

[19] Katia Lupinetti, *Identification of shape and structural characteristics in assembly models for retrieval applications*. Dissertation, École Nationale Supérieure d'Arts et Métiers, Paris, 24.01.2018.

[20] J. R. Reich, "Ontological design patterns: Metadata of molecular biological ontologies, information and knowledge," in *Database and Expert Systems Applications*, pp. 698–709, Springer Berlin Heidelberg, 2000.