INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.70

Knowledge Documentation Based on Automatic Identification and Clustering of Change Intentions in CAD Data of Wiring Harnesses

K. Eder [1,2,✉], W. Herzog [1], M. M. Altner [1,3], O. Tas [1], J. Neckenich [1] and K. Paetzold [2]

[1] Mercedes-Benz AG, Germany, [2] Technische Universität Dresden, Germany,
[3] Karlsruhe Institute of Technology, Germany

✉ kevin.eder@mercedes-benz.com

Abstract

High amount of changes and increasing complexity in CAD design of wiring harnesses result in a lack of time for documentation and transfer of acquired knowledge. To be able to transfer the gained knowledge efficiently during development automating the identification, analyzation and documentation of changes is necessary. This paper shows a methodology to address this challenge for CAD data of wiring harnesses. Thus, it is shown how interrelated change elements can be combined or separated from each other according to their change intention.

Keywords: design knowledge, knowledge management, knowledge representations, computer-aided design (CAD)

1. Introduction

As digitalization proceeds, the need for high quality data increases. One key area is the design of 3D CAD (computer-aided design) models. CAD data are forming the basis of many downstream processes as simulations, assembly planning and drawing less manufacturing release (Neckenich, 2017). Since the development time of products is continuously decreasing, the efficiency of design processes has to be increased. This in particular challenges the automotive wiring harness development. Due to the high dependence on the surrounding components of the wiring harness, changes have to be carried out daily (Smith, 2015). Because of their high complexity, these modifications mainly require a particularly high effort (Kuhn and Nguyen, 2019). In addition, overlapping timelines and subsequent changes to the wiring harness cause changes to be made to various data sets (Eder *et al.*, 2022).

In order to address these challenges, knowledge-based engineering (KBE) applications are an appropriate solution. These aim to reuse existing knowledge by automating individual design tasks and thus reduce development time (Bracewell *et al.*, 2009). Since knowledge can only be acquired through the interpretation of existing information, it has to be extracted and processed according to the particular use case (Chandrasegaran *et al.*, 2013). As each product and task has its own unique challenges, KBE applications are standalone solutions and require their dedicated knowledge management (KM) (Gorski *et al.*, 2016; Lundin *et al.*, 2017).

To address the challenges of the wiring harness development (Eder *et al.*, 2022) describe an approach for a corresponding knowledge management system (KMS) consisting of four stages (see Figure 1). In the first stage (1. Knowledge acquisition) design changes are made to a wiring harness CAD model. Afterwards the two sequential data sets are compared with each other in order to identify and document the changes being made (2. Knowledge documentation). Based on this knowledge package, it is possible to identify potential reusers in a database (3. Knowledge transfer) and subsequently integrate the knowledge (4. Knowledge reuse). Since several changes with various change intentions can be made from one version of a wiring harness to another, each change information must be clustered corresponding to its intention. For example one connector was added in one area of the wiring harness and the routing has been changed in another area. Consequently, these can be considered as single independent change intentions. This enables the transfer of separated changes and increases reusability due to the reduced complexity. Furthermore, each individual change is enriched by an automatically generated change description, which is used to inform potential reusers about the changes that have been made.

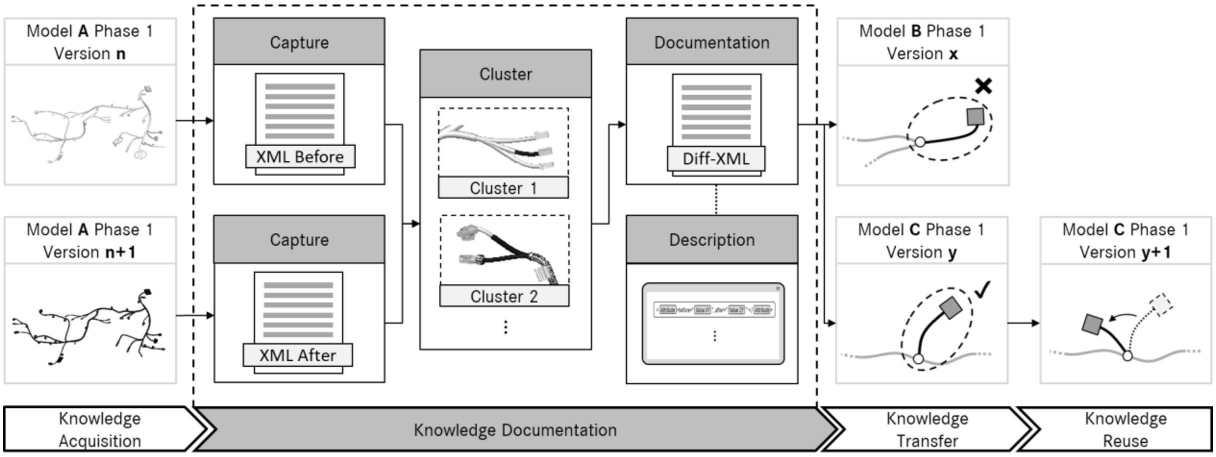

Figure 1. KMS for wiring harness development on the basis of (Eder *et al.*, 2022)

To advance the stage of the knowledge documentation of the KMS approach the following questions have to be answered:

- How can changes in CAD data of wiring harnesses be captured automatically?
- How can related changes be clustered and separated from each other according to their change intention?
- How can this knowledge be documented in order to integrate it into other CAD data sets?
- How can a description of the changes be generated automatically?

This paper is organized as follows: Section 2 gives an overview of the state of the art of knowledge documentation for CAD Data. Section 3 describes the approach to capture changes in CAD Data of wiring harnesses automatically, cluster them according to their design intention and enrich them via change description. The shown approach is validated in Section 4. Finally Section 5 concludes and presents an outlook on future research.

2. State of the art

There are many methodologies available to guide the development of KBE systems. One of the best known is the methodology for knowledge-based engineering applications (MOKA) (Stokes, 2001). Core aspects of MOKA are the capturing and formalization of knowledge. Capturing is the process of converting the relevant information into a structured knowledge representation (KR) that is readable as well as processable by machines. The objective of formalizing is the preparation of knowledge for subsequent processes and making it comprehensible for users (Verhagen *et al.*, 2012).

2.1. Knowledge representation of CAD data

CAD data are built up from information about their structure, topology, relations, geometry and product manufacturing information (PMI) (Cho *et al.*, 2017). However, these data are encrypted, allowing this information only to be interpreted by the corresponding CAD system itself. Nevertheless, these systems allow user-defined information to be exported via API (application programming interface) into a neutral format, such as XML (extensible markup language) (Colombo *et al.*, 2014).

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

For the wiring harness, there is such a standardized object-oriented exchange format based on the XML schema in form of the KBL (wiring harness list) (prostep ivip and Verband der Automobilindustrie, 2018). This format is used to document, amongst other things, information about geometric parameters, used components, topologies, as well as references to other documents. The wire paths are described by B-spline curves and therefore can differ from the user-defined points in the CAD system. For example, in commercial CAD applications like Siemens NX the user can define spline paths using pole points or through points. Nonetheless, comparing two sequential KBL data sets could thus theoretically reveal all change scenarios. However, when the information is subsequently reintegrated into a CAD dataset, the B-Spline Curve would have to be converted, resulting in deviations from the original design. Furthermore, subsequent adjustments can be complicated because of the number and position of the reinterpreted routing points, which can differ significantly from the original design.

Another standardized exchange format for the wiring harness is the VEC (Vehicle Electrical Container) (prostep ivip and Verband der Automobilindustrie, 2020). This is also based on the XML scheme. But in contrast to the KBL format, it contains information about the entire electronic system of a vehicle. Nevertheless, the individual spline curves are also not described by the user-defined routing points, but by Non-Uniform rational B-Splines (NURBS) curves. Thus, this format would also lead to deviations from the original design during reintegration into the CAD system.

In addition, there are numerous other neutral exchange formats that can be used for KBE applications for CAD data sets. One of the most popular is the STEP (Standard for the Exchange of Product model data) format, standardized by ISO 10303 (International Organization for Standardization, 2020). This is mainly used for the exchange between two different CAD systems. The disadvantages of the STEP format, however, is the lack of information of the design history, parameters, constraints and features of the original CAD model (Kim *et al.*, 2008). This also makes it unsuitable for recognition and clustering according to the design intent of the wiring harness.

2.2. Design process knowledge capturing

The information required to carry out operations between individual stages of the product lifecycle are described as design process knowledge (Chandrasegaran *et al.*, 2013; Hoisl *et al.*, 2008). One method to capture this is presented in the works of (Sung *et al.*, 2009; Sung *et al.*, 2012). In this case, the commands executed by the user are recorded and documented in an XML file. As an alternative, design process knowledge can be recorded by comparison of the input state with the output state based on the same structure (Dworschak *et al.*, 2021). Depending on the use case and chosen format of the knowledge representation, customized comparison methods have to be used. One method to compare changes in XML files is presented by (Oliveira *et al.*, 2020). In their approach, changes to individual attributes are not only detected, but also semantically analyzed within the overall context using pre-defined rules and accordingly summarized.

2.3. Design intent documentation

In order to be able to reuse knowledge, increasing the understandability by transferring the design intention is essential (Verhagen *et al.*, 2012; Camba *et al.*, 2014). Thus, captured knowledge must be enriched in such a way that it can be understood quickly and easily by everyone (Ball *et al.*, 2006). For instance, the work of Ritchie *et al.*, 2008; Sung *et al.*, 2009 shows how the design intent can be converted from XML data into an English syntax, thus improving the understandability of the content.

Nevertheless, there is still a research gap when it comes to the inclusion of the design intent in KBE applications (Cho *et al.*, 2016). In general, capturing the design intent requires additional effort during data acquisition. In Sung *et al.*, 2012, for example, the designer is questioned about the respective reason for his executed action during the design process. Other approaches such as the one by Camba *et al.*, 2014 are based on the use of annotations in the CAD model. A major barrier in capturing the design intent is that these approaches require additional effort on behalf of the creator, who may refuse to invest additional time (Szykman *et al.*, 2001). Moreover, it should not be underestimated that the additional effort reduces the overall efficiency of the KBE process, which significantly influences the acceptance and success of a KBE application (Lundin *et al.*, 2017; Dworschak *et al.*, 2021).

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

3. Knowledge documentation in wiring harness development

The main task of the wiring harness development in CAD, is the collision-free routing and connection of the components. Most changes to the wiring harness become necessary as result of adjustments to the surrounding vehicle components (Smith, 2015). For example, if an electronic control unit is moved, the position of the associated connector and the corresponding routing of the wiring harness have to be adjusted. Since the change behavior of the component itself is irrelevant for the KMS of the wiring harness, the change intention implicitly results from the changes being made.

In order to be able to automatically identify changes between two version stages and cluster them according to the respective design intention, the steps from Figure 2 are carried out. First, the general structure of a wiring harness is analyzed. Based on this, all relevant information is identified, which is necessary for the exact description of a wiring harness model in CAD. In order to enable the processing of this knowledge, a knowledge representation is specified using an XML scheme documenting all the relevant information. Subsequently, all change scenarios are defined, which can be carried out for each individual element. These are then linked to the information contained in the XML files. Based on this, it is analyzed which dependencies exist between the individual changes and thus rules are defined to cluster related ones. Subsequently, it is determined how the processed knowledge could be documented for future integration into other data sets. Finally, text modules are defined, which can be compiled into change descriptions automatically using the previously documented knowledge.

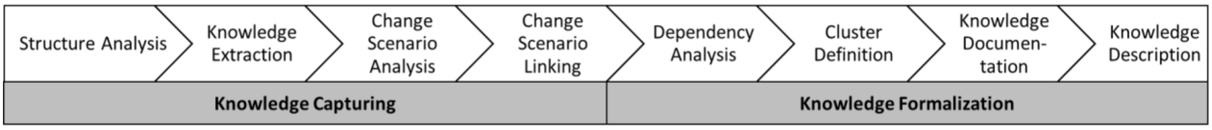

Figure 2. Processing steps enabling automatic identification and clustering of change intention

3.1. Structure of a wiring harness

The purpose of a wiring harness is to connect the individual electrical components of a vehicle, such as control units, actuators and sensors. To facilitate the handling of the wiring harness, the individual wires are bundled. This results in a tree-like structure consisting of many individual segments. The individual segments can be differentiated into two types. End segments describe the last section of a node up to a connector and mid-segments describe the section between two nodes. In order to bundle individual wires and protect them against environmental influences, they are provided with a wire protection. In order to mount the wiring harness to the vehicle, fixing elements as well as accessories such as labels can be attached to the segments (see Figure 3).

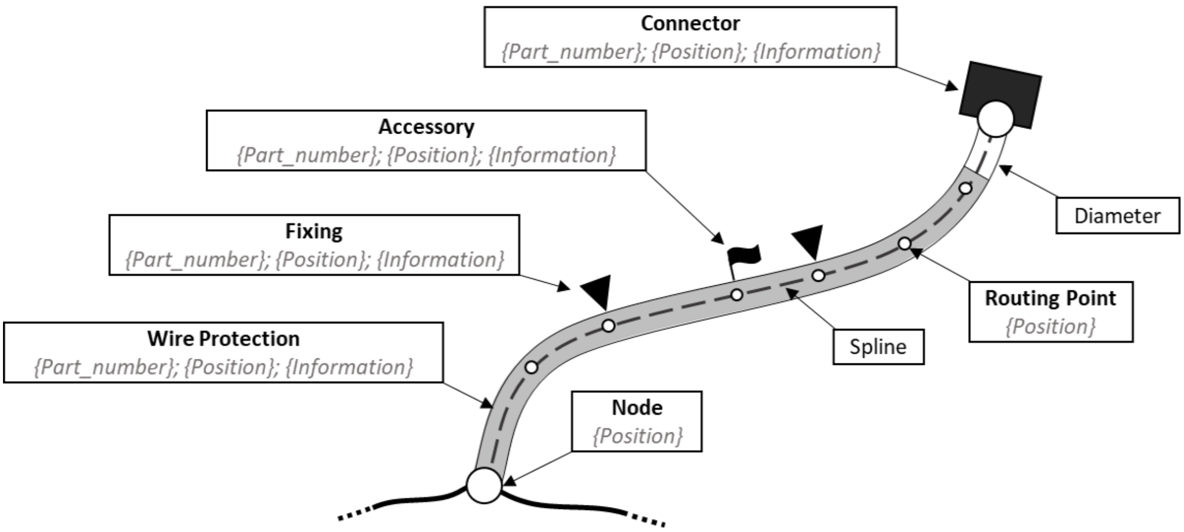

Figure 3. Structure and elements of a wiring harness

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

3.2. Knowledge representation of CAD data of wiring harnesses

As each Wiring Harness can differ depending on the selected equipment configuration, the CAD model represents the maximum space reservation in the vehicle. Thus, each segment defines the largest possible bundle diameter and does not contain any information of single wires. In general, the space reservation includes the wire protection and manufacturing tolerances. (Neckenich *et al.*, 2016).

A wiring harness assembly is composed of several parts. The central part contains all segments including wire protections and accessories. All required assembly parts, such as connectors, and fixings are added separately. Since the CAD model of a wiring harness is limited to a specific installation space, the wire bundles are designed following the top-down principle in the global coordinate system of the complete vehicle (Neckenich *et al.*, 2016).

The basic skeleton of a segment is a spline (see Figure 3). Its path is defined by the coordinate values of the individual routing points and nodes. Vectors at the respective nodes define the direction of the branch off. Using sweep representation along the path of the spline the maximum diameter of the wiring bundle is represented. Furthermore, segments can be provided with PMI, for instance, about length tolerances or material properties. Connectors and fixings are modelled according to the bottom-up principle as separate CAD models in their own local coordinate system and provided as parts by the PDM system (Product Data Management) (Neckenich *et al.*, 2016). Since the main aspect of the wiring harness development in CAD is the routing and connecting of the components, the change management of the individual parts is not relevant for this approach. As both connectors and fixings are imported from the PDM system, information about the part number and the positioning data (location and orientation) in relation to the global coordinate system are required. These parts are always tangentially linked to the segments. Hence, the documentation of the assembly constraints can be omitted. To enable the CAD model to be linked to the circuit diagram, a unique reference is required for any connector indicating its destiny, for example, that a specific connector is linked to the control unit of the left front seat.

Depending on the used CAD system, wire protections and accessories can be represented using PMI on the segment or using parameterized reusable objects. For the latter, the respective product or part number of the reuse library is required. Wire protections can be described by start and end locations relative to the segment. Accessories are defined by coordinates and their orientation. In order to make the extracted knowledge easier to understand for users in subsequent processes, additional implicit attributes are exported. Some examples are element descriptions or the lengths of segment and wire protections.

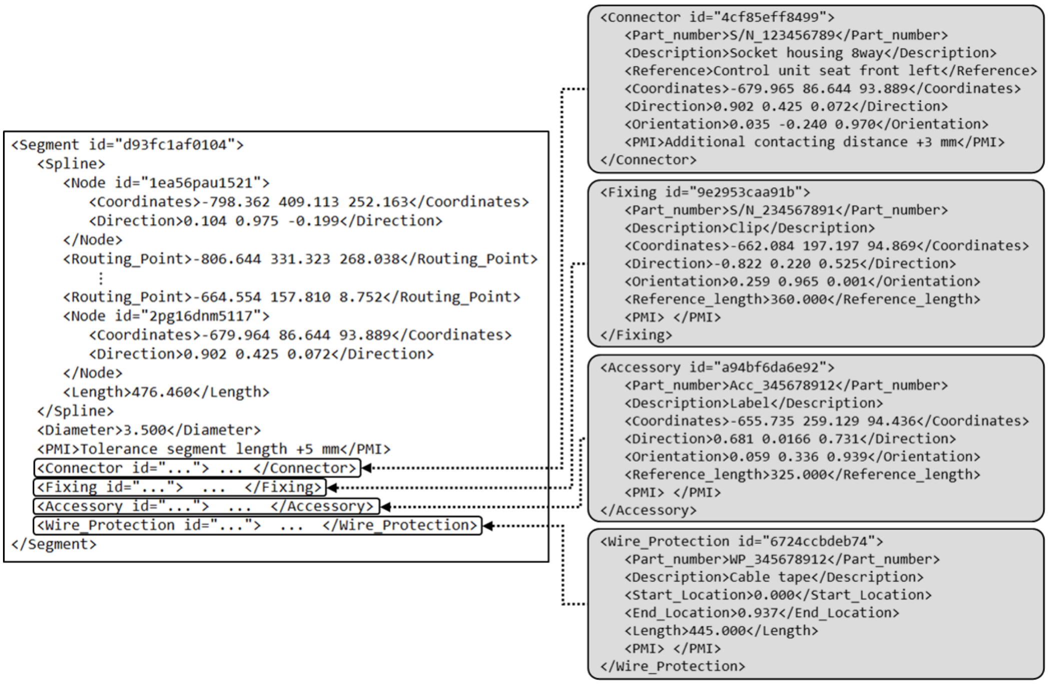

Figure 4. Knowledge representation of a wiring harness segment in XML scheme

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

In order to identify each element used during the comparison of two data sets, all elements should be provided with their own unique identification number (ID) (Neckenich, 2017). This ID has to be generated in the CAD application when a part is added to the assembly or once a segment is created. Figure 4 shows how the knowledge representation of a Wiring Harness Segment in XML format can look like. In contrast to the KBL and VEC format, the user-defined routing points are used. In order to consolidate all information more comprehensively, an object-oriented structure was not applied. This helps in order to ensure both comparability and reintegration into subsequent processes.

3.3. Identification of change scenarios

Modifications in CAD can be divided into the basic categories add, delete and change. The latter can be subdivided into geometric and informal changes. In addition, the operator replace, as well as the operators split and join are used for splines respectively segments. These are actually combinations of the basic change types mentioned above. For example, the replace operation is nothing more than the deletion and subsequent constrained addition of an element. Splitting a segment in turn means adding a new segment and at the same time adjusting the original segment. Joining a segment is the exact contrary of this. Especially for the transfer and reuse of knowledge, these constraints are essential for the functionality and additionally increase the efficiency of the whole system. The implied constraint of these operators prevents users from deleting a connector without adding its replacement connector. In order to be able to recognize the individual change scenarios, it is essential to know which attributes involve which type of information. Based on that individual change scenarios can be linked to the individual attributes of the knowledge representation (see Figure 5).

+ = New   o = Changed
- = Missing   • = Possibly Changed

		Segment									Wire Protection						Connector					Fixing					Accessory				
		Add	Delete	Replace	Split	Join	Change Routing	Change Length	Change Diameter	Change Information	Add	Delete	Replace	Change Position	Change Information	Change Length	Add	Delete	Replace	Change Position	Change Information	Add	Delete	Replace	Change Position	Change Information	Add	Delete	Replace	Change Position	Change Information
Part Data	ID	+	-	o	o	o					+	-	o				+	-	o			+	-	o			+	-	o		
	Part Number										+	-	o				+	-	o			+	-	o			+	-	o		
Information	Description										+	-	o		o		+	-	o		o	+	-	o		o	+	-	o		o
	Reference																+	-	•		o										
	PMI	+	-	o	o	o				o	+	-	•		o		+	-	•		o	+	-	•		o	+	-	•		o
Geometry	Node [Coordinates]	+	-	o	o	o	o	•																							
	Node [Direction]	+	-	o	o	o	o	•																							
	Routing Point [Coordinates]	+	-	o	o	o	o	•																							
	Coordinates																+	-	•	o		+	-	•	o		+	-	•	o	
	Direction																+	-	•	o		+	-	•	o		+	-	•	o	
	Orientation																+	-	•	o		+	-	•	o		+	-	•	o	
	Start Location										+	-	•	o		o															
	End Location										+	-	•	o		o															
	Diameter	+	-	o	o	o			o																						
Implicit	Length	+	-	o	o	o	•	o			+	-	•	•		o						+	-	•	•		+	-	•	•	

Figure 5. Linking of the change scenarios to the attributes of the wiring harness

Using the unique ID of the elements, it is possible to detect whether they have been added, deleted or adapted when comparing two versions. In order to be able to recognize whether the changes are a replace, join or split, additional rules must be applied. For example, if a wire protection, connector, fixing or accessory is deleted from a segment and a new one is added, this is classified as a replacement. Another example is, if a segment is deleted and a new one is connected to the same connector, this must also be classified as a replacement. Adding a new node always involves splitting a segment. The affected segments are thus identifiable via the new node. When joining, the affected segment is identifiable by the two nodes, which in the previous state were at the respective end of two adjacent segments.

3.4. Clustering of change scenarios

The objective of clustering changes is to separate independent changes from each other and bundle related changes in order to prevent changes from being made only partially, as in the case of a replacement. The

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

smaller the cluster, the higher the probability that the respective changes can be integrated into other data sets. To achieve this, changes that occur globally across the whole wiring harness have to be separated from each and clusters must be formed.
Based on the link between the individual change scenarios and the individual attributes, the changes related to a segment can be identified and assigned. Thus, a segment that contains all changed as well as unchanged data forms the smallest possible cluster. The latter are required in subsequent processes for matching purposes. In many cases, changes are not limited to one segment, but are related to other segments. For example, if a node is added, removed or moved, several segments are affected by this modification. Thus, for the documentation of the change intention, it is essential to identify the related changes and to combine them in a local cluster. Since cross-segment changes always occur in combination with modifications to nodes, these can be identified on the basis of the ID and coordinates of the nodes and therefore summarized accordingly.
Theoretically, local clusters that are combined based on these rules can consist of several change intentions. In Figure 6, for example, a connector P1 was moved. At the same time, a new segment A3 including a new connector P2 was added to the associated segment S1, causing it to split into A1 and A2. Therefore, these could be two change requests executed simultaneously by the designer. Thus, the separation of these changes into two element-based cluster would increase the probability of a potential reuse. However, in order to document the respective changes separately, there is a lack of information. Since the path of segment B1 is a combination of segment A1 and segment A2, the changes of delta B could be documented completely. For delta C, on the other hand, there is no user-defined position for node CN and no routing path as well as a defined length for segment C3. The separate documentation of delta B and delta C would require the documentation of delta A, since the position of node AN, as well as the routing path and the length of segment A3, is not included in delta B or delta C.

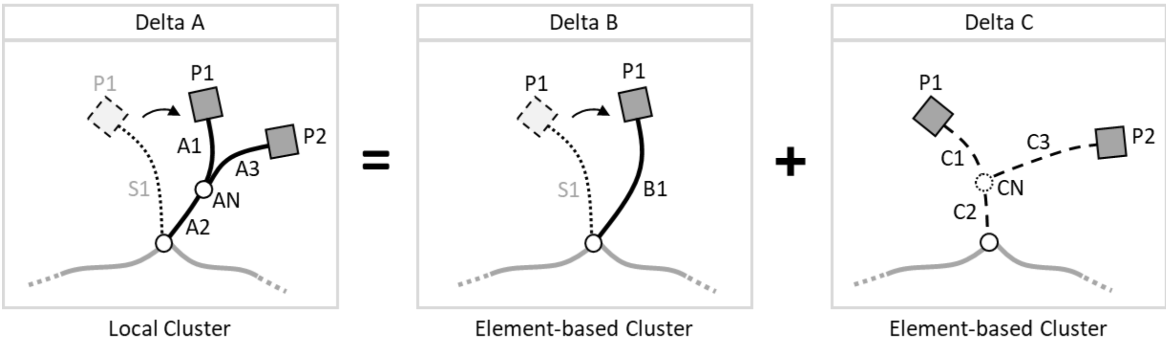

Figure 6. Exemplary subdivision of a local cluster into element-based cluster

To reduce the file size, the knowledge transfer system should rather be enabled to extract element-based knowledge from documented local cluster. Thus, possible reusers could be informed about the respective change intentions. The missing information would then have to be complemented by the reuser himself.

3.5. Knowledge documentation

The knowledge documentation serves to store the gained knowledge in a way that possible reusers can be identified in subsequent processes automatically. Furthermore, this knowledge package should include all information in order to integrate it into other data sets afterwards. For this reason, a XML scheme (Diff-XML) based on the knowledge representation of section 3.2 inspired by (Oliveira *et al.*, 2020) was specified. In this the content of the previous and the new state are compared with each other (see Figure 7). In order to perform matching with other data sets and additionally enable further extractions of knowledge, as shown in Section 3.4, all information about a segment is documented. All changes of a cluster are documented in a corresponding subtree and thus separated from others.

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

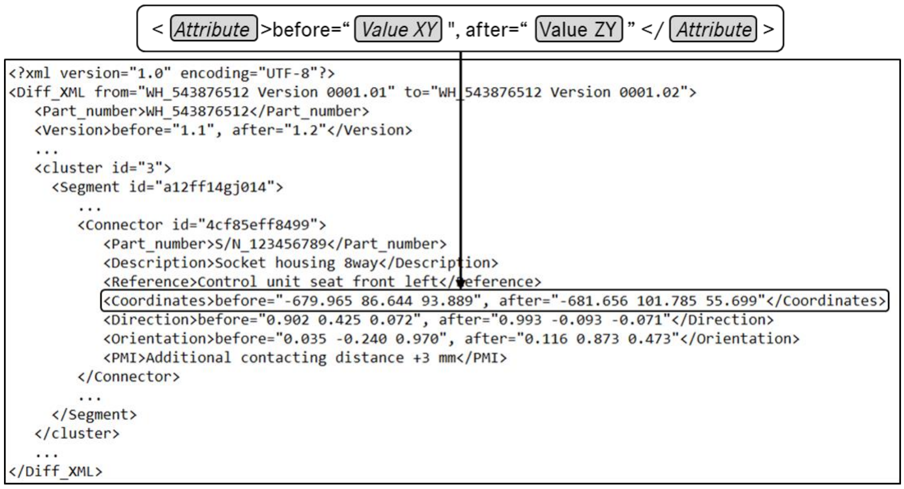

Figure 7. Exemplary excerpt of a Diff-XML

3.6. Knowledge description

In many KMS, the design intention has to be documented by the designer with additional effort reducing the user acceptance and the efficiency of the entire system. Since changes to the wiring harness in CAD are typically consequences of changes to the surrounding components, the change intention results implicitly from the changes made. Thus, standardized change descriptions can be generated automatically using text modules for all possible change scenarios (see Figure 8). These modules are descriptions of the affected element, the change operator and the before/after state. The change descriptions can be store in the Diff-XML file. Alternatively, they could be compiled and presented to the user by the system during the reuse process. The latter leads to a reduced file size of each individual Diff-XML, but requires more computational effort in subsequent processes.

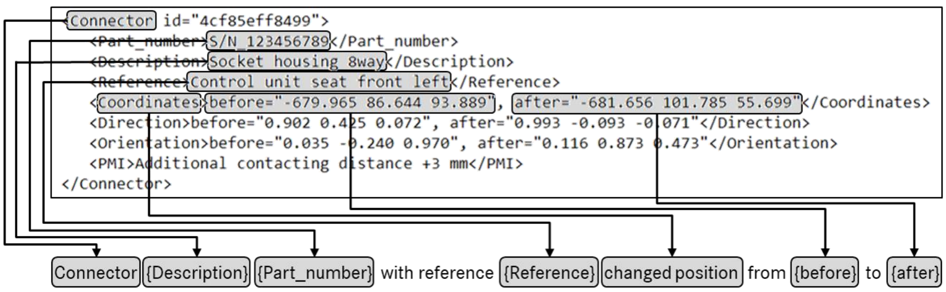

Figure 8. Exemplary text modules for automatically generated change descriptions

4. Validation

The methodology was validated using a CAD model of a wiring harness in which different and globally distributed changes were made (see Figure 9). Those were chosen so every change operation and every element was used at least once. To capture the initial and final state, one XML file was derived via API before and one after the changes were made. To identify all changes, each segment and their corresponding elements were checked. If a segment or element is new, missing or any attribute has changed, the after and before state of the whole segment was transferred into a Diff-XML file as one cluster as shown in Figure 7. For example, in cluster three the position of the connector and the corresponding segment routing has changed. Thus, the length of the segment and wire protection has changed as well. Since the start node was not modified the cluster is limited to that one segment.

If a change of a node occurred to more than one segment the corresponding segments were combined in one cluster. For example in cluster 2, where an existing segment was split due to a new added segment. Thus all change scenarios could be identified and clustered accordingly. In addition, the corresponding change descriptions for each cluster could be generated. Therefore, it could be shown that the presented methodology works successfully.

https://doi.org/10.1017/pds.2022.70 Published online by Cambridge University Press

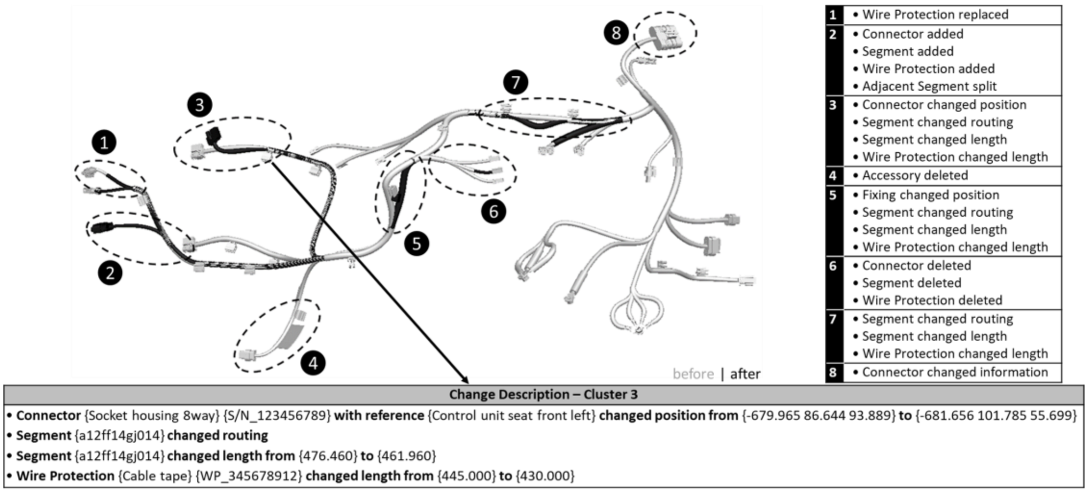

Figure 9. Validation model with globally distributed changes and exemplary change description

5. Conclusion and outlook

Due to concurrent engineering, changes in wiring harness development have to be integrated repetitively into multiple data sets sometimes. The transfer and reuse of knowledge is one solution to address this. Therefore, knowledge have to be captured and formalized first. Thus, we have presented a methodology to capture and cluster changes in CAD data sets of wiring harnesses automatically. The clusters are supposed to separate independent changes from each other and increase the probability for their reuse. At first, an XML-based knowledge representation was specified containing all relevant information of a wiring harness CAD model. Rules were defined to identify all change scenarios and cluster them comparing two XML files. Furthermore, it was presented how the knowledge could be documented to support its further reuse. In addition, it was shown how this knowledge can be used to generate change descriptions automatically using predefined text modules. Thus, making the contained knowledge more comprehensible. Finally, the methodology was validated with an exemplary wiring harness CAD model. Future work will investigate how this knowledge documentation can be used to detect potential reusers automatically and how the information can be integrated into other CAD data sets. This should include an investigation of the utilization of partial information from the local clusters. Subsequently, experienced wiring harness designer should test the whole system in order to evaluate its usability and its efficiency. Furthermore, it can be considered to use the generated change descriptions to analyze the change behavior in wiring harness development. Beyond that, transferring this methodology to the design process of other products that rely on a modular structure could be investigated.

References

Ball, A., Patel, M., McMahon, C., Culley, S., Green, S. and Clarkson, J. (2006), “A Grand Challenge: Immortal Information and Through Life Knowledge Management (KIM)”, *The International Journal of Digital Curation*, 2006

Bracewell, R., Wallace, K., Moss, M. and Knott, D. (2009), “Capturing design rationale”, *Computer-Aided Design*, Vol. 41 No. 3, pp. 173–186. https://doi.org/10.1016/j.cad.2008.10.005

Camba, J., Contero, M., Johnson, M. and Company, P. (2014), “Extended 3D annotations as a new mechanism to explicitly communicate geometric design intent and increase CAD model reusability”, *Computer-Aided Design*, Vol. 57, pp. 61–73. https://doi.org/10.1016/j.cad.2014.07.001

Chandrasegaran, S.K., Ramani, K., Sriram, R.D., Horváth, I., Bernard, A., Harik, R.F. and Gao, W. (2013), “The evolution, challenges, and future of knowledge representation in product design systems”, *Computer-Aided Design*, Vol. 45 No. 2, pp. 204–228. https://doi.org/10.1016/j.cad.2012.08.006

Cho, J., Vosgien, T. and Gerhard, D. (2017), “Engineering Knowledge Extraction for Semantic Interoperability Between CAD, KBE and PLM Systems”, Vol. 517, pp. 568–579. https://doi.org/10.1007/978-3-319-72905-3_50

Cho, J., Vosgien, T., Prante, T. and Gerhard, D. (2016), *KBE-PLM Integration Schema for Engineering Knowledge Re-use and Design Automation*, Vol. 492, Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-54660-5

Colombo, G., Pugliese, D., Klein, P. and Lutzemnberger, J. (2014), "A study for neutral format to exchange and reuse engineering knowledge in KBE applications", 2014 *International Conference on Engineering, Technology and Innovation (ICE)*, pp. 1–10. https://doi.org/10.1109/ICE.2014.6871565

Dworschak, F., Kügler, P., Schleich, B. and Wartzack, S. (2021), "Model and Knowledge Representation for the Reuse of Design Process Knowledge Supporting Design Automation in Mass Customization", *Applied Sciences*, Vol. 11 No. 21, p. 9825. https://doi.org/10.3390/app11219825

Eder, K., Tas, O., Zielbauer, U. and Paetzold, K. (2022), "A Knowledge Management Approach to Support Concurrent Engineering in Wiring Harness Development", in Canciglieri Junior, O., Noël, F., Rivest, L. and Bouras, A. (Eds.), *Product Lifecycle Management. Green and Blue Technologies to Support Smart and Sustainable Organizations*, *IFIP Advances in Information and Communication Technology*, Vol. 640, Springer International Publishing, Cham, pp. 68–79. https://doi.org/10.1007/978-3-030-94399-8_6

Gorski, F., Zawadazki, P. and Hamrol, A. (2016), "Knowledge based engineering as a condition of effective mass production of configurable products by design automation", *Journal of Machine Engineering, Vol. 16, No. 4*

Hoisl, F., Shea, K. and Helms, B. (2008), "Towards representing, evolving and networking engineering knowledge for computational design synthesis", *The 10th International Design Conference on Proceedings DESIGN 2008 Dubrovnik, Croatia*

International Organization for Standardization (ISO) (2020), ISO 10303-242: Industrial automation systems and integration - Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering

Kim, J., Pratt, M.J., Iyer, R.G. and Sriram, R.D. (2008), "Standardized data exchange of CAD models with design intent", *Computer-Aided Design*, Vol. 40 No. 7, pp. 760–777. https://doi.org/10.1016/j.cad.2007.06.014

Kuhn, M. and Nguyen, H. (2019), "The future of harness development and manufacturing - Results from an expert case study", *Bordnetze im Automobil - 7. Internationaler Fachkongress*, 2019

Lundin, M., Lejon, E., Dagman, A., Näsström, M. and Jeppsson, P. (2017), "Efficient Design Module Capture and Representation for Product Family Reuse", *Journal of Computing and Information Science in Engineering*, Vol. 17 No. 3. https://doi.org/10.1115/1.4035673

Neckenich, J. (2017), "3D-Master-Leitungssatz–Konzept zur Entwicklung von Leitungssätzen als 3D-Master in einem realistischen, vollständigen DMU-Modell"

Neckenich, J., Zielbauer, U., Winter, R. and Vielhaber, M. (2016), "An integrated approach for an extended assembly-oriented design of automotive wiring harness using 3D Master Models", *DS 84: Proceedings of the DESIGN 2016 14th International Design Conference*

Oliveira, A., Kohwalter, T., Kalinowski, M., Murta, L. and Braganholo, V. (2020), "XChange: A semantic diff approach for XML documents", *Information Systems*, Vol. 94, p. 101610. https://doi.org/10.1016/j.is.2020.101610

prostep ivip and Verband der Automobilindustrie (VDA) (2018), "prostep ivip VDA Recommendation: Harness Description List (KBL) Version 2.5"

prostep ivip and Verband der Automobilindustrie (VDA) (2020), "Vehicle Electric Container (VEC). Version 1.2"

Ritchie, J.M., Sung, R.C.W., Rea, H.J., Lim, T., Corney, J.R., Salamon, C. and Howley, I. (2008), "Automated Knowledge Capture in 2D and 3D Design Environments", *2nd International Workshop Virtual Manufacturing VirMan 08 as part of the 5th INTUITION International Conference: Virtual Reality in Industry and Society: From Research to Application*

Smith, N. (2015), "Enhancing Automotive Wire Harness Manufacturing Through Digital Continuity", *SAE 2015 World Congress & Exhibition*. https://doi.org/10.4271/2015-01-0238.

Stokes, M. (Ed.) (2001), Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications, London, Bury St. Edmunds

Sung, R.C., Ritchie, J.M., Robinson, G., Day, P.N., Corney, J.R. and Lim, T. (2009), "Automated design process modelling and analysis using immersive virtual reality", *Computer-Aided Design*, Vol. 41 No. 12, pp. 1082–1094. https://doi.org/10.1016/j.cad.2009.09.006

Sung, R.C.W., Ritchie, J.M., Lim, T. and Kosmadoudi, Z. (2012), "Automated generation of engineering rationale, knowledge and intent representations during the product life cycle", *Virtual Reality*, Vol. 16 No. 1, pp. 69–85. https://doi.org/10.1007/s10055-011-0196-8

Szykman, S., Sriram, R.D. and Regli, W.C. (2001), "The Role of Knowledge in Next-generation Product Development Systems", *Journal of Computing and Information Science in Engineering*, Vol. 1 No. 1, pp. 3–11. https://doi.org/10.1115/1.1344238

Verhagen, W.J., Bermell-Garcia, P., van Dijk, R.E. and Curran, R. (2012), "A critical review of Knowledge-Based Engineering: An identification of research challenges", *Advanced Engineering Informatics*, Vol. 26 No. 1, pp. 5–15. https://doi.org/10.1016/j.aei.2011.06.004