



13th CIRP conference on Computer Aided Tolerancing

Automated point-based tolerance analysis model creation for sheet metal parts

F. Litwa^{a*}, M. Gottwald^a, M. Bohn^a, J. F. Klinger^a, M. Walter^b, S. Wartzack^b, M. Vielhaber^c^aDaimler AG, Benzstraße, 71059 Sindelfingen, Germany^bInstitute of Engineering Design, Friedrich-Alexander-University Erlangen-Nuremberg, Martensstrasse 9, 91058 Erlangen, Germany^cInstitute of Engineering Design, Saarland University, Germany

* Corresponding author. Tel.: +49-7031-9078210 ; fax: +49-711-3052136639. E-mail address: Frank.Litwa@daimler.com

Abstract

This paper focuses on a concept that shows a way to automatically create a point-based tolerance analysis model out of existing development data. Nowadays solutions for an automated tolerance simulation model creation are using a static approach for the model build-up. For this purpose product-/ production- development data are automatically mapped on preexisting models (e.g. skeleton models). If changes during development process occur, the tolerance simulation models have to be reworked. Today only simple changes in the model can be automated (e.g. change of distribution, tolerance range etc.). A complete new tolerance simulation model build-up process for dynamically changing product-/ production- development information is not possible. To give an application example, tolerance simulation models for sheet metal parts in automotive industry are based on different development data. Before the first simulation model is created (to secure the tolerance concepts etc.), all necessary information have already been developed, e.g. in the automotive industry's development process: part geometry, tolerance information, assembly graph, jig and fixture concept, joining location and measurement points. Thus the automated simulation generation should be possible.

First step is to describe an interface for a dynamic model creation in tolerance simulation systems. In a second step preprocessing of development data is necessary to map them into tolerance simulation software restrictions. This delivers a solution to fill the gap between the PDM-/ CAD and the CAT-system. The considered approach for automated tolerance simulation model creation provides the opportunity to build-up the tolerance analysis models highly efficient and almost automatically. Tolerance analysis can then be used to rapidly calculate several options. This offers the possibility to increase the product maturity level at a very early stage of the development process.

© 2015 The Authors. Published by Elsevier B.V This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the organizing committee of 13th CIRP conference on Computer Aided Tolerancing

Keywords: tolerance simulation; automation; development process; interface; methodology

1. Introduction

In today's mass production environment each component of a product contains deviations. This leads to functional as well as aesthetical losses. This is one reason why today hardware prototype phases are existing to secure the interaction between several components. The shortening of the development period for new products causes a shifting from the hardware prototype phases into the digital CAX-world. To prevent undesirable deviations of assemblies already in an early stage of product-/ production-development where no physical data exists, tolerance analysis software is an indispensable tool. With the aid of these analyses it is possible to calculate between which limits single parts are allowed to deviate, so that the functionality of the assembly is guaranteed. Therefore it is necessary to model

the whole assembly process of the parts in the manufacturing plant in tolerance simulation software. Thus the build-up of such a tolerance analysis model requires several pieces of information. This data can be divided into product development data, such as part geometry and tolerance information, production development data, such as assembly graphs, jig and fixture concepts, joining locations and measurement points.

Large scale enterprises store these data in a PDM-system. Modern PDM-systems provide the possibility for centralized data management through all stages of development process. But the focus of today's PDM-systems is in organizing product development data. For product-/ and production-development related data (e.g. assembly graph for several vehicle configurations, production locations) there are no adequate solutions available [1]. For this reason, the

production development related data is often stored in a format that is not digital utilizable in follow-up processes.

Nevertheless all these information are necessary to define contact conditions, tolerances and measurements in tolerance simulation software. Nowadays these information are gathered and a manual build-up of the tolerance simulation model is performed based on practical tolerance simulation knowledge (figure 1).

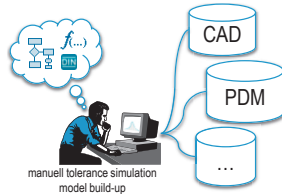


Fig. 1 Manual process to build-up tolerance simulation models

This manual process is time consuming and fault-prone. For instance, the build-up process for a tolerance analysis in the automotive industry takes several days / weeks (depending on the scope of the set of data to be verified).

As mentioned in [3, 4] simulation tools are indispensable to decrease the development time. Furthermore, in an early development stage, the cycle for updates in product- and production- development data is short termed [5]. For that reason, the results of tolerance analysis are often based on obsolete data what complicates decision-making. This is why in practice tolerance simulation tools are used to build-up a small scope of data instead of modelling e.g. the whole car. As a result it is very difficult to recognize complex relationships where e.g. single part tolerances are affecting the final product negatively. Hence in today's market situation, such a working process is not acceptable anymore.

An efficient automated tolerance simulation model build-up out of digital existing product- and production-development data seems to be a solution.

This paper introduces an approach to create point based tolerance simulation almost automatically. Therefore chapter 2 gives a short overview of existing interfaces in CAT-software to simplify the tolerance simulation model build-up process. Also the fundamentals in designing an interface are presented. Chapter 3 introduces an approach of a neutral exchange formats' architecture to transfer product- / production- development data to tolerance simulation software. Furthermore an abstract algorithm is shown to map the development data in CAT-specific structure. Chapter 4 draws a conclusion of the presented approach and gives an outlook on future research opportunities.

2. State of the art and deficits

In general CAx- systems do provide interfaces using standardized data format for exchange between different systems. Often these exchange formats fulfill their requirements and work satisfactorily. For special fields of applications (e.g. CAT), these standardized exchange formats are inefficient. This means there is a loss of information from one system to the other [6] as shown in figure 2.

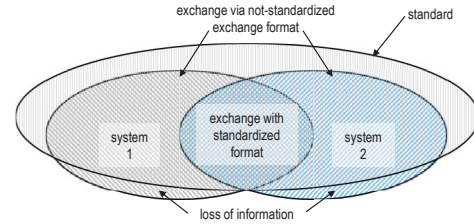


Fig. 2 General problem using standardized exchange formats based on [6]

The following chapter shall point out, that modern CAT-software does not fulfill the requirements for an automated built-up process, even if all the necessary data to create tolerance simulation models are available in the development process.

2.1. Existing interfaces in CAT-software

In order to simplify the generation process of simulation models current CAT-systems do provide several possibilities using non-standardized exchange formats. For example there are interfaces which allow mapping tolerance information, measurements and moves on preexisting model (MTM) [7]. Disadvantage of this solution is the creation of the assembly structure in the simulation environment before. The whole modeling process is time consuming. After the manual build-up the information for tolerance simulation has to be mapped manually on the assembly structure. Thus, a complete automated model generation using this interface is not possible.

Another possibility to create tolerance simulation models automatically is the usage of Excel interfaces [7]. As presented in [8] this interface allows it to e.g. build-up a specific CAT-model out of a standardized parametric CAD-model using specific engineering data as exact geometry, tolerance information, contact conditions and measurements from different sources. Disadvantage of this approach is the static assembly structure of the model has to be known. The parametric model is static and does not allow changes in the assembly sequence (insert / delete parts, change in the assembly sequence) caused by dynamically changing product- / or production- development information. During the development process of different types of cars or derivate of one car, there are often changes in the assembly sequence or even different assembly sequences (considering different manufacturing plants for the same car). Thus, to grant high quality standards, it is necessary to create tolerance simulation models on each car type or derivate on each iteration of development process. This is the only way to give an adequate feedback if changes in geometry, assembly sequence etc. are necessary to achieve the objective.

Therefore it is essential to develop an interface which allows an automated completely independent build-up of a tolerance simulation model out of existing product- / production- development data. The focus of attention must be a solution with as less as possible human interaction during the build-up process. Instead of the manual process the usage of tested algorithm is reasonable to avoid mistakes.

2.2. Fundamentals of interface designing

As mentioned before, one focus of this paper is the specification of an interface from PDM / CAD-system to CAT-software. Therefore the basic method for defining an interface between two systems will be shown. The core of the interface is a system independent format which carries all the necessary information from the one system to the other. The architecture of this system independent format is subjected to several specifications. On the one hand e.g. the existing development structure of the product-/ production-development data is giving specifications. On the other hand there are specifications from e.g. the tolerance simulation environment so that the system is able to handle the information.

If architectural communalities are existing regarding the specifications of the system independent format (1 in figure 3) and if there are analogies in the functional dependencies of the data (2 in figure 3), it is possible to grant dataflow through an interface. This means, the development data can be restructured in a neutral data format (with the help of an algorithm) so that the information can be reused from tolerance simulation software.

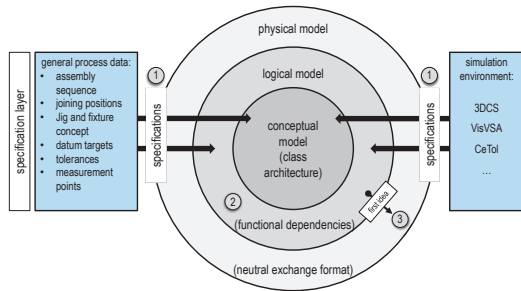


Fig. 3 Process for developing a software independent exchange format (specification process) based on [9]

This gives a first idea of the system independent exchange formats architecture (3 in figure 3). The whole process is called “specification process”.

After the specifications were brought in from the parties involved (customers development process; tolerance software environment), there has to be made an agreement. This agreement shall define which party is responsible to implement an interface in the specific system, based on all the specifications gathered (4 in figure 4).

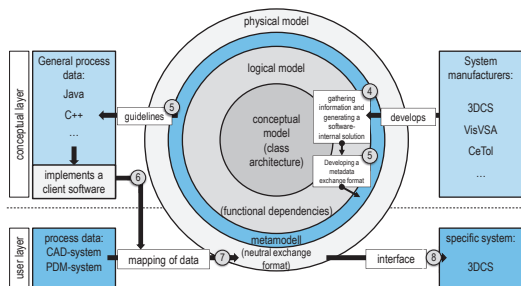


Fig. 4 Workflow to define an interface between two systems (conceptual layer) containing the system specific solutions (user layer) based on [9]

Having the new interface implemented, the architecture has to be given in a metamodel format to the other participating party (5 in figure 4). This party can then start to ensure the information read out of the development data is restructured by the help of an algorithm into the right architectural structure based on the provided metamodel (of the tolerance simulation environment) (6 in figure 4).

By completing the implementation of the application for restructuring the data, it is then possible to built-up an abstract tolerance simulation model in a neutral data format (7 in figure 4) which can be imported into specific tolerance simulation software (8 in figure 4).

This basic method to define an interface between two software solutions shall now be elaborated for tolerance simulation purpose.

3. Approach for tolerance simulation software

To ensure that the concept is working, it is necessary to export product-/ production- development information from an existing PDM-/ CAD-system. For an automated export of data modern PDM-/ CAD-systems do provide a common application programming interface [10] which can be used. Usually these interfaces are based on an object-orientated language. To implement a solution for the system in a specific programming language, the API often offers different language support [10]. Due to the API it is possible to access public data or subprograms in the required system.

As mentioned before the two parties (customers development process; tolerance software environment) are both giving restrictions. Fulfilling them ensures that the exchange of information is possible. As follows the restrictions from the development process and CAT-software are shown.

3.1.1. Dependency of information in the development process

From product-/ production- development process point of view there are relations and dependencies between the different kinds of information. As an example these relations and dependencies are demonstrated with the development process in automotive industry as shown in figure 5.

In the beginning of development process the external design of the product is created. Based on this data, it is possible to start with the construction of several other parts underneath the external cover. Also the product management decides how e.g. gaps run at the external cover. This decision is based on models out of claim and virtual models. End of the design process are released data (strak) and also a detailed gap- and radii plan in a first step (1 in figure 5).

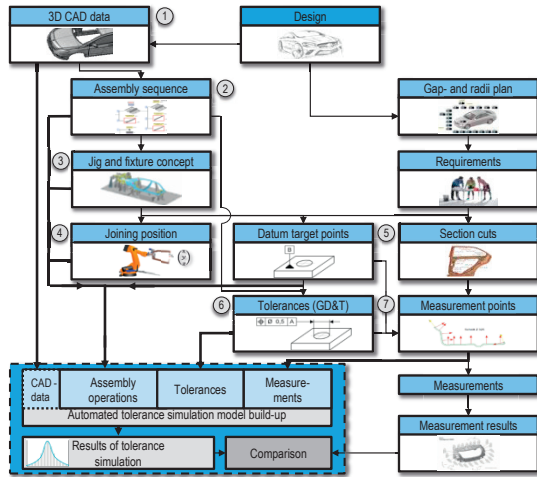


Fig. 5 Relations and dependencies between the development data and their usage in tolerance simulation software [11]

Next step is the construction of the internal car structure. During this process there also is basically decided how the assembly sequence shall look like in the manufacturing plant (2 in figure 5). Based on the assembly sequence it is possible to develop the jig and fixture concept for each assembly step (3 in figure 5).

Knowing the jig and fixture points of the parts in the plant, it is also possible to decide which positions can be used to e.g. weld the parts together. Normally welding operation for connecting parts is always performed between two tighteners (4 in figure 5). As mentioned in [12, 13] datum targets (5 in figure 5) are often defined at position where:

- elastic deformation of the part (e.g. due to gravity) is on a minimum,
- process capability of the part can be guaranteed,
- wearing of the part during manufacturing process is on a minimum.

For geometric elements of the part which are not used as datums but set up a function, tolerances have to be defined (6 in figure 5) [13].

Regarding quality management, the requirement is, each jig and fixture point has to be either measured or is used as a measurement datum target point for mounting the part on the measuring machine. Therefore a section cut is used to define either measurement datum target points or measurement points (7 in figure 5). For measurement datum target points the usage of datum target points is common practice. Also the restrictions from the gap- and radii plan have to be checked. Thus, section cuts also exist on each gap of the gap- and radii plan. Out of the product- and production- information (1-7 in figure 5) the build-up of tolerance simulation models is performed.

3.1.2. Derived requirements for a neutral exchange format (from development process point of view)

Considering this background information a neutral exchange format should be structured as follows (Figure 6).

Not every class is exactly defined in this discourse. This paper shall only give an overview for basically required classes.

At first there has to be implemented the hierarchical part structure which represents the assembly sequence. This means a structure which shows the relationships between parts and assemblies. One possibility in realizing this structure is to implement a rooted hierarchical tree structure. In case of the assembly sequence the order is limited due to the finite depth of the structure. As an example every *Assembly*-class in this structure could consist out a fixed number of attributes such as *Assembly name*. Another attribute in the *Assembly*-class may refer to classes like *Part* or *Subassembly*.

The *Subassembly*-class is necessary to store assemblies of suppliers and for storing the jig and fixture concept. This class provides an attribute *Assembly* and is able to point towards one / several assemblies. This enables the hierarchical assembly structure.

The *Part*-class contains several attributes like *Part name*, *Path* (where the geometry is stored), *Geometries* (part describing geometry), *Datum targets and Tolerances*. For point based tolerance simulation the geometry is not needed, but helps for a correct result interpretation.

The attribute *Geometries* is able to contain several other instances of classes like *Surface*, *Edge*, *Vertice* or *Point*, etc. Then again these classes are detailed via attributes like *ID*, *Curvature*, *Direction*, etc.

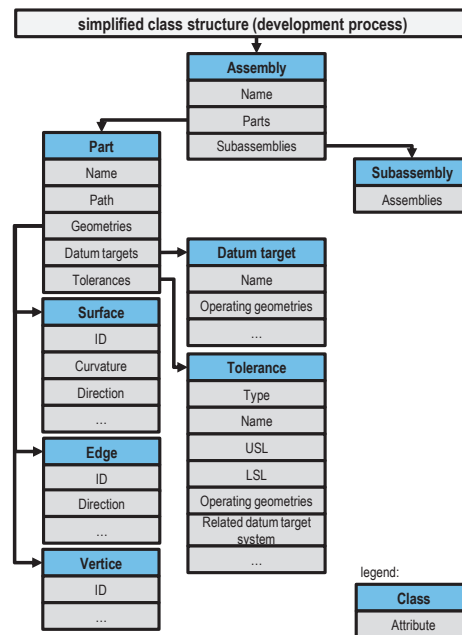


Fig. 6 Structure (simplified class structure) of development data due to restrictions of the development process

3.1.3. Dependencies of information in CAT-software

From CAT-software point of view there also are relations and dependencies between the different kinds of information.

Figure 7 shows the standard workflow to build-up a tolerance simulation model [14].

In the beginning the CAD-data of the scope of data to be verified is brought to the specific CAT-software (1 in figure 7). As mentioned before the exact geometry is not needed to build-up a point based tolerance analysis model, but it helps for correct interpretation of the simulation results. In the next step the assembly operations have to be defined so that the parts are assembled correctly according to the assembly sequence (2 in figure 7). For surfaces of parts that are involved in the tolerance chain and are not used for defining a datum target system, tolerance information is defined in the development process. If points on these surfaces are used to set up assembly operations, the tolerance information has to deposit at these assembling points (3 in figure 7). The next step is to define measurement points for the quality features of the product (4 in figure 7). To start the tolerance simulation it is also necessary to set up specific tolerance simulation parameters. Based on this data, it is possible to start the tolerance simulation.

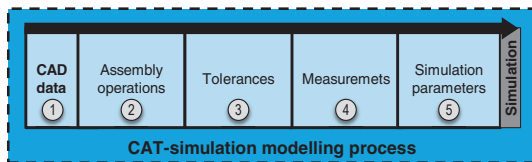


Fig. 7 Relations and dependencies in CAT-software (simulation model build-up process)

Bearing in mind this background information the tolerance simulation software itself gives restrictions to the structure of a tolerance simulation model.

3.1.4. Derived requirements for a neutral exchange format (from CAT-software)

For point based tolerance simulation models (frequently performed in body in white automotive industry) these requirements can be seen in figure 8.

The top level is defined by the *Simulation*-class. One attribute of this class is *Name* which specifies the simulation name. This class also contains attributes like *Parameter*, *Assembly* which are referring to the corresponding classes. The *Parameter*-class includes all parameters which are necessary for tolerance simulation (e.g. number of Monte-Carlo runs, GD&T standard ...).

The *Assembly*-class consists out of attributes like *Name*, *Path*, *Parts*, *Subassemblies*, *Moves* and *Measurements*. Here also the *Subassembly*-class has an attribute *Assembly* which is able to point on several classes of *Assembly*. Thus the hierarchical assembly structure can be created.

The *Tolerance*-class for example is an abstract class which carries attributes like e.g. *USL*, *LSL*. These standard attributes are used for every derived tolerance-class, which can be completed with attributes like *Name*, *Related datum target system* or *Angle*.

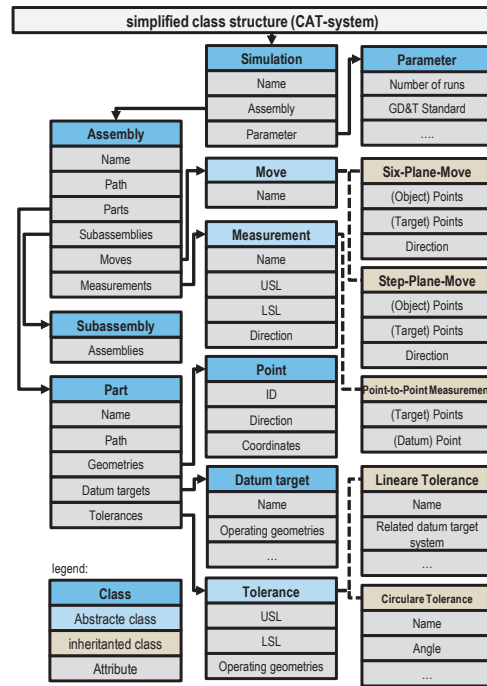


Fig. 8 Structure (simplified class structure) of point based tolerance simulation data due to restrictions of tolerance simulation software

3.2. General algorithm to map requirements from development data to CAT-requirements

As mentioned in chapter 2 the exchange of data through an interface can only be performed if there are architectural commonalities existing (taking the restrictions of chapter 3.1.1 and 3.1.3 into account) and analogies found in the functional dependencies of the information. By comparing figure 6 and figure 8 analogies can easily be detected. For example, both systems use a hierarchical tree structure to model the assembly sequence. Also product and production information are stored in parts and assemblies. The storage of assembly operations and the definition of measurements seem to be the grave differences. Due to the functional dependencies of development data it is possible to assign the correct information to measurements and assembly operations. In these cases an algorithm ensures the correct mapping of development data into the tolerance software specific requirement. For all the other development data it should be easy to get them into tolerance simulation software structure.

For example the mapping of data which shall be used for a specific tolerance simulation model can be done in eight steps as shown in figure 9.

First step is to create the required class structure for each part and assembly (used for the specific simulation) shown in figure 8.

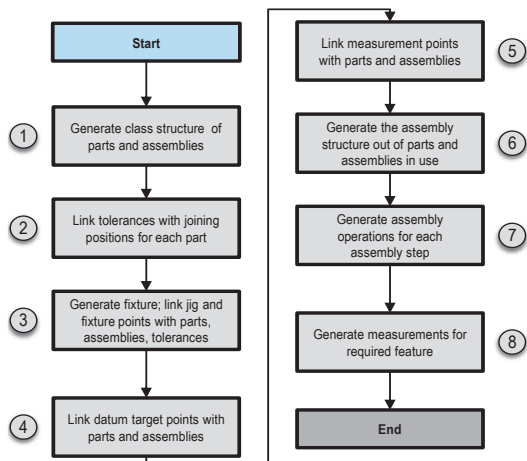


Fig. 9 Mapping of development data for tolerance simulation format [11]

In a second step each joining location (point) has to be saved in the related parts. Also these locations have to be linked with part tolerances for this position (e.g. specific part tolerances for the surface which is used for joining process or general part tolerances).

Saving the jig and fixture points in the associated parts is the third step. These points also have to be linked with the specific surface tolerances (of the parts). To model the manufacturing plant in the tolerance simulation environment, the jig and fixture points are used. These points are stored in a new part (fixture) which has the same class structure as the standard parts. This new fixture-part ensures manufacturing plant tolerances can be displayed beside simple part tolerances. These tolerances are important because often the positioning process of parts and assemblies in the manufacturing plant leads to serious deviations in the final assembly. The fourth step is to assign datum target points to the related parts. Matching measurement points (using the right tolerance range: USL, LSL) with the specific part or assembly is the fifth step. An important step to create the specific tolerance simulation model is the implementation of the assembly sequence. Here the parts and assemblies are organized in a specific structure (e.g. in a XML format). Linking the parts with the help of assembly operations is the 7th step. These assembly operations are equipped with specific points (joining points, jig and fixture points, and datum target points). An algorithm has to ensure the correct choice of points. Last step is to define the measurement for measuring the quality feature.

Finally the restructured development data can be saved in a neutral data format (e.g. XML) and can be imported in tolerance simulation software which provides the interface specified in chapter 2.

4. Conclusion

The concept for an automated point based tolerance simulation model creation shown in this paper allows generating a tolerance simulation model with a minimum of human interaction.

With the help of the automated model build-up process it is possible to investigate dimensional accuracy of a series production process for instance in automotive engineering on each iteration of development process. It is so possible to give a well-founded feedback to e.g. construction engineers for eventual geometric changes of parts. Also the automated build-up process allows calculating several versions for e.g. alignment of parts / assemblies in the manufacturing plant. Thus the optimal tolerance concept can be found to ensure the assembly process of a product leads to a high quality product. This shows an idea to successfully establish tolerance simulation in the development process and to increase the product maturity level at a very early stage of the development process.

Future work involves developing an approach to store product- and production- development data in a uniting PDM-system. This is the only way to ensure digital consistency of simulation relevant data in the development process.

Also the generation of a complex algorithm to ensure correct mapping of development data into tolerance simulation requirements e.g. for definition of contact condition (moves in tolerance simulation model) should be given more attention.

References

- [1] VDI 2219, "Informationsverarbeitung in der Produktentwicklung – Einführung in die Wirtschaftlichkeit von EDM/PDM-Systemen" VDI, Düsseldorf, 2002.
- [2] Zhou Y., Vivek T., Monteverde V., Morgan E.: Feature-based and Point-based Variation Modeling. Dimensional Control Systems, Inc. 2013
- [3] Backhaus R. Porsche Panamera: 16% weniger Entwicklungszeit dank Simulation. <http://www.atzonline.de/Aktuell/Nachrichten/1/8819/Porsche-Panamera-16-Prozent-weniger-Entwicklungszeit-dank-Simulation.html> [published: 2008-11-28; last accessed 2013-12-30]
- [4] Zerrweck F. Qualität bei Mercedes-Benz – vom Einzelteil zum Gesamtfahrzeug. Presentation. Mercedes-Event Center, 2011-07-15
- [5] Ehlenpiel, K.: Integrierte Produktentwicklung. 4. Auflage. München: Carl, Hanser Verlag, 2009 – ISBN 978-3-446-42013-7.
- [6] Engelken, G., Wagner, W.: Unigraphics – Praktikum mit NX5. 2. Auflage. Wiesbaden: Springer Science, 2008 - ISBN 978-3-8348-0408-3
- [7] Dimensional Control Systems, Inc: Software Help Manual. Version 6.19.0.0.0; 1994-2010
- [8] Watrin, S., Gottwald, M., Binz, H., Xu, X., 2012. "Three-dimensional parametric CAT-model for sensitivity analysis of rear axle drives" p. 18.
- [9] Jeckle, M.: Entwurf von XML-Sprachen. <http://www.jeckle.de/entwurfxml/entwurfxml.html> [published: 2004-06-08; last accessed 2013-12-30]
- [10] Siemens PLM Software: NX programming and customization – Extending and tailoring NX digital product development solutions. http://www.plm.automation.siemens.com/de_de/products/nx/for-design/productivity-tools/customization-programming.shtml [published: 2010; last accessed 2013-12-30]
- [11] Litwa, F: Entwicklung eines Konzepts zur automatisierten Erstellung von Toleranzsimulationsmodellen aus vorhandenen Produkt- und Prozessdaten. Masterthesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2013
- [12] Jordan, W.; Schütte, W.: Form- und Lagetoleranzen. Handbuch für Studium und Praxis. 7. Auflage. München: Carl Hanser Verlag 2012 - ISBN 978-3446430808.
- [13] Bohn, M.; Hetsch, K.: Toleranzmanagement im Automobilbau. 1. Auflage. München: Carl Hanser Verlag GmbH & Co. KG, 2013. – ISBN 978-3-446-43496-7
- [14] Variation Analysis Systems: Software Help Manual. Version Teamcenter 9.1; 2011