Management and Production Engineering Review
Volume 8 • Number 2 • June 2017 • pp. 91–98
DOI: 10.1515/mper-2017-0021

DE G DE GRUYTER OPEN

CONCEPT OF PRODUCT COMPLEXITY MODELING IN THE DEVELOPMENT OF A MACHINE SHAFT

Jerzy Pokojski, Łukasz Woźnicki

Institute of Machine Design Fundamentals, Warsaw University of Technology, Poland

Corresponding author:
Jerzy Pokojski
Warsaw University of Technology
Institute of Machine Design Fundamentals
Narbutta 84, 02-524 Warsaw, Poland
phone: (+48) 22 234-82-48
e-mail: jerzy.pokojski@simr.pw.edu.pl

Received: 18 November 2015
Accepted: 27 December 2016

Abstract

In the paper product and design process modeling on the basis of speed reducer is presented. The paper proposes to build the product and process models and parallel to that carry out the project activities as these models reflect the detailed structure of the projects.
The most important feature of the proposed approach is its possibility to model only important parts of the models and to keep track of the development of their chronological paths. The approach may be treated as a partial CPM/MOKA model or as network-integrated CPM/MOKA models.

Keywords

product complexity modeling, object oriented modeling, UML.

Introduction

At present many products are designed and manufactured as standardized complex structures (for instance automotive components: elements of braking systems, seats, control systems, steering systems, etc.) [1–5]. They appear in numerous variants which differ in structure, parameters, way of design, manufacturing etc. Each series of these products can vary in several of the features mentioned.

Products like these are usually made by one of the component suppliers which operate in world markets and constantly optimize their product profiles [1, 6, 7]. The suppliers are very specialized in their areas of functioning. In a closer look at such products, it becomes obvious that a lot of engineering knowledge is behind their production. The manufacturers collect knowledge not only dealing with the form and structure of the components [2, 6, 8] but also with the details of the product models and the associated design processes.

The variety of certain product states and configurations is nowadays very high [2, 9]. It has become standard to store information about the way the products are built or how a particular example of the product is configured. It is also very important how this product was designed, what was calculated (and how), analyzed and simulated.

There are software systems which can store all that information [2–5, 9, 10]. But the fact that the established information structures evolve continuously makes it a difficult task as it is not easy to operate with such changing information. Very careful computer modeling – both of the designed product and the design process – is an essential predisposition to resolve the matter.

The product/process model structure has to fit for the components and their mutual assemblies, associations, and relationships which are included to the complete and complex product description. Engineering processes require various and numerous analyses, simulations and real-life examinations and are the result of a step by step evolution of the respective product. The design process in its single steps can be made on a very standard and routine way but it can also be performed as a very innovative action. The whole structure should offer the possibility to store dynamically developing product/process models regarding their real life, industrial, and engineering aspects.

These models can then be associated with the classic, standard documentation of the project. Each product component placed in the documentation can be also classified and integrated with its, usually object-oriented, model.

Looking at performed real-life engineering processes, it becomes obvious that innovations may appear at different stages of engineering activities. (Usually, the remaining tasks are more static and routine.) In order to store information concerning the designed product and the design process, it is necessary to concentrate on those components which are new and innovative, because the others are well known and standard.

Comparing similar engineering processes (in case of different projects) – even when performed in the same design office – differences in general issues as well as with some sophisticated details can be spotted.

To analyze the documentation of a project it is very important to recognize the new elements of the product or its process.

Comparing the results achieved in different projects can also help to select those projects which have novel and valuable elements in their documentation.

The proposed solution of the product complexity modeling and further considerations are shown using the example of a speed reducer (and its shafts) design.

The objectives of the paper are: 1) to model on the basis of Core Product Model (CPM) [3] and MOKA (Methodology and tools Oriented to Knowledge-based engineering Applications) [10]) the changes of core product/process models which have an innovative character, 2) to use the concept of the base model and to modify and develop it later (in the case if a feature is not given in the actual model, it is assumed that it remains the same as in the previous model).

Concept and exemplary application

Introduction

The paper proposes to build the product and process models and parallel to that carry out the project activities as these models reflect the detailed structure of the projects.

For the development of the speed reducer [8] two different approaches were considered which apply: CPM [3] and MOKA [10] as design product/process models.

The Core Product Model (CPM) was developed at National Institute of Standard and Technology [3–5]. It is a generic model expressed as a UML class diagram. The CPM concentrates on three aspects: function, form, behaviour. The offered set of models allows to build conceptual, intermediate and implementation models.

Initially, the CPM model was designed as a universal representation, which is expandable and independent of any product development. Over some years the CPM has been corrected, developed and specialised.

Finally, a number of formalisms with its details and structures was created, for instance for engineering design: description of geometry, function, form, bahaviour, material, different physical, functional decompositions, mappings, realtionships.

Additionally other similar formalisms were developed. The MOKA project (the effect of ESPRIT project [10]) also uses UML standard models. The MOKA supports the life-cycle of Knowledge Based Engineering (KBE). The MOKA approach bases on informal model and formal model [10]. The informal models use ICARE forms. ICARE means Illustration, Constraint, Activity, Rule, Entiety. These forms are used as knowledge representations for many user perspectives. Later this knowledge is transformed into a so called formal (structured) model which can be used for building KBE applications. The formal models are based on MML (MOKA Modeling Language which is a kind of UML representation [10]). The MOKA consists of a limited number of sequentially performed steps. The most important are the two steps: “capture” and “formalize”.

Examples made on the basis of MOKA are mostly product models. The CPM was developed as a neutral representation for catching product development information which can be used in the cooperation with exisiting and future engineering systems. The MOKA is a formalization to develop more efficient KBE application.

At first glance, both options seem to be suitable for the considered class of applications. But there are also some deeper and more sophisticated differences. With the CPM a very detailed product model based on UML and its structures can be built. When applying the MOKA, first engineering knowledge is stored and then an informal model is created (knowledge is acquired from engineers [10]). This means that this approach does not offer special tools for universal solutions. It starts with knowledge collecting and then builds an informal model. After the informal model has been created it is transferred into a formal model expressed as MML model. Of course, it is possible to create at once formal models on the basis of information from engineers. In this case there is no need to develop a very precise object oriented structure with the complete product/process modeling and filling it with all possible detailed data. It is only necessary to establish such structures which reflect the key knowledge structures human designers regard essential. Consequently, only details which are important (or should be taken into account) in given case, can be modeled and there is no need to cover each detail. Hence, such models/structures can be detailed only in interesting fields and areas while others can contain only general information.

The role of the CPM [3–5] is to provide support for product data use, storage, communication during product life cycle. The CPM offers possibilities to realize all the above functionalities for a certain product.

The CPM consists of a set of components [3–5]: abstract classes (CoreProductModel, CommonCoreRelationship, CommonCoreObject, CoreEntity, CoreProperty), object classes (Artifact, Feature, Port, Function, TransferFunction, Form, Geometry, Material, Flow, Behaviour, Specification, Requirement), relationship classes (Constraint, EntityAssociation, Usage, Trace), utility classes (Information, ProcessInformation, Rationale). Associations and Aggregations also are provided. The CPM models have 3 levels: conceptual, intermediate, implementation. There are also developed extensions of the CPM model: Open Assembly Model, Product Family Evolution Model [5] and others. The CPM allows to store information from different stages of the product development. Each stage inherits information from the previous stages and prepares information for the next stages.

Papers concerning the CPM and their examples of engineering product/process modeling show rather limited and simplified real life objects. The paper [1] is an exception. But it also shows huge work and effort connected with the modeling of the complete set of information concerning considered class of objects – planetary gear-boxes.

The CPM formalism seems to be a sensible and practical solution. But the attempt of applying it for products like an automotive combustion engine, an automotive suspension or an automotive seat (and others) with its real, industrial and full data sets is very complicated and expensive.

On the other hand the world market automotive suppliers develop many products which are nearly the same, they are variants of products based on identical solutions. Sometimes there are complete product families. With such cases the information stored in the CPM model would be often the same, containing banal, standard, usually well-known information. A model like that is in general rarely changed and improved. It can be created once for a longer period of time and be called the base model of the certain product.

But parallel to the above characteristics there are also core changes performed in some details of engineering processes (for instance the details of Finite Element Method analysis of automotive seats [7, 11]) which are worth collecting and storing with the help of local components of the CPM model. The stored models may be very similar to each other in some parts though completely different in others.

The authors tried to exploit this concept with a speed reducer which was treated as the reference model of the automotive seat.

The concept is illustrated with the example of a speed reducer (and its shafts) which was developed over some period of time.

Example

The stored information concerns 3 stages of design product/ process development.

At each phase the design process is realized by human designers using computer tools. The computer tools are developing from stage to stage. In principle, at each stage the same design process of the same product is performed. But when looking closer it can be noticed that it is done differently and different computer tools are used. Parallel to that information about the particular case is stored. This information is captured in design product/process models.

At each stage a geometric model of the speed reducer is generated (which is created) with the help of a CAD system. But operations performed by the human designer are different. With every design some kind of progress is noticeable. The implementation of new tools makes the whole process faster and more automatic. The information about the performed processes and achieved product models is stored in accompanying models.

The development of a speed reducer design processes development can be separated into the following stages:

1. The conceptual design of the speed reducer and its basic calculations are made by the designer by hand. Computer tools are only used for the generation of the geometric model with the help of Knowledge Based Engineering tools [8].
2. The design process of the speed reducer is supported by a computer program which works according

to the linear design process model [8]. In 8 steps different parts of the speed reducer and their parameters are selected. Some are taken from classical catalogues, others from computer data bases. Computer tools are only used at the end of the process to generate a geometric model with the help of Knowledge Based Engineering tools.

3. The design process is supported by computer tools which allow to model the forms of the shafts automatically.

The 3 different stages are presented in Fig. 1. It also shows the product models which were used in each case.

Figure 2 contains snapshots from different software systems used for designing the shafts belonging to the speed reducer. Figure 3 illustrates the shaft product models and the respective design processes. The core elements of the speed reducer product model are shown there. For the process of the machine shaft design, treated as a part of the speed reducer, more elaborate models are depicted.

In the presented proposal it is assumed that only core elements belong to the product/process models. Missing information has to be taken from earlier models. The whole data structure of the product/process models development is built as a network of certain chronologically realized design cases (Fig. 4). The chronology links for the product and process models are built and performed separately.

The information stored in the proposed data structure offers the possibility of comparing two cases according to different definitions. For instance: a) all elements are the same, b) some prescribed elements are the same – some not, c) some selected elements are present, etc. It is recommended to follow the hierarchic order when comparing several elements.

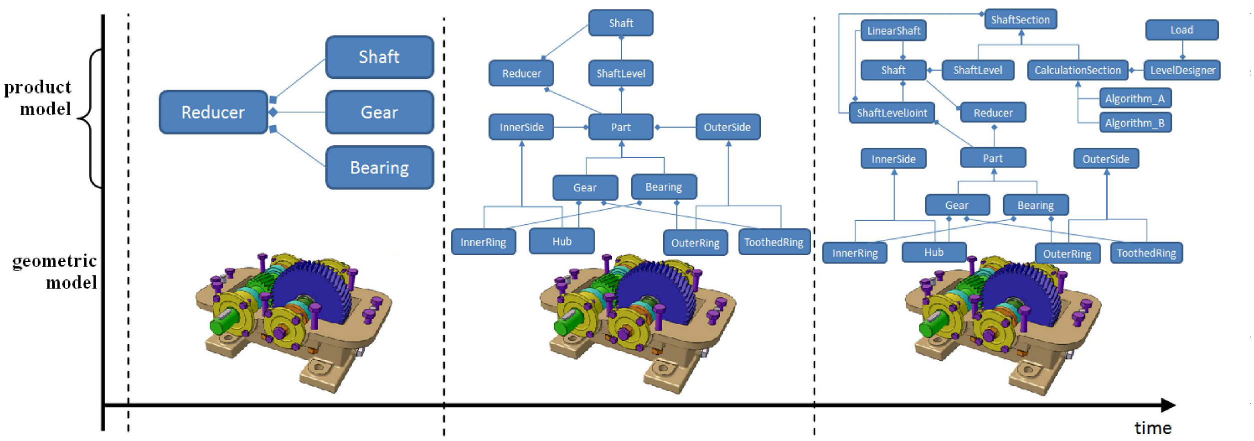

Fig. 1. Exemplary geometric models of speed reducer and their product models.

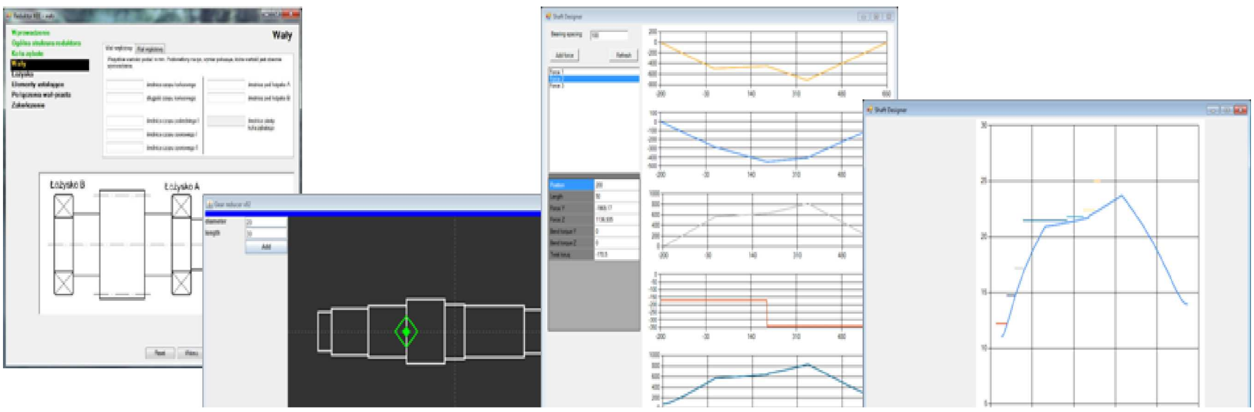

Fig. 2. Snapshots of the software supporting the design process of shafts.

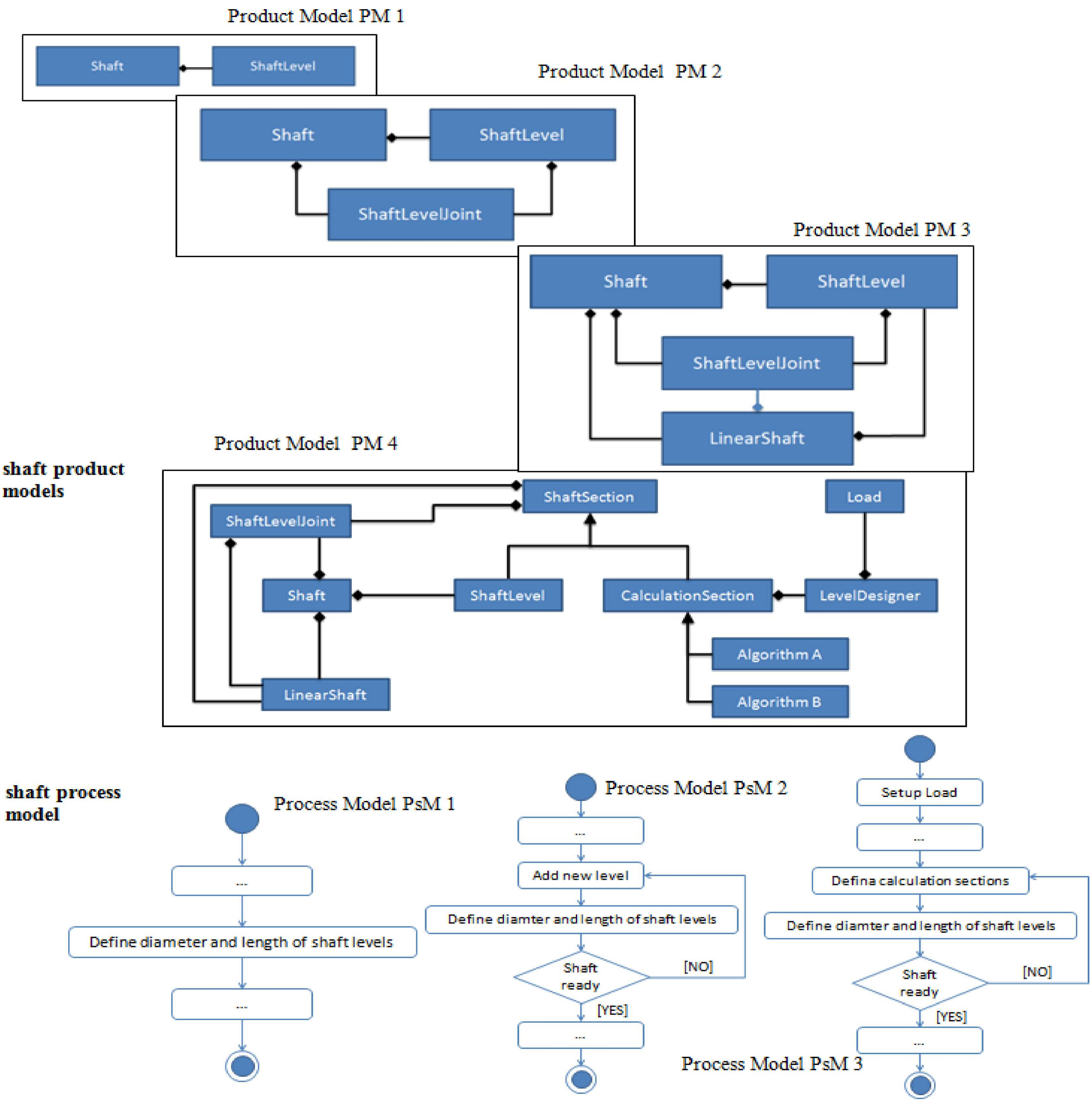

Fig. 3. Shaft – product and design process models.

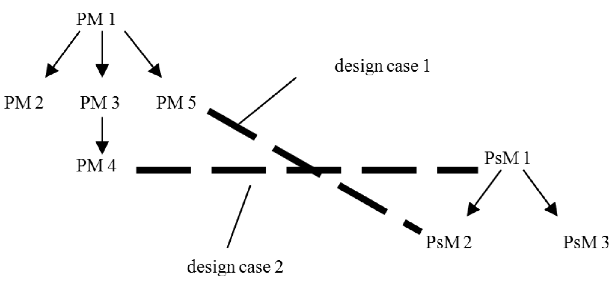

Fig. 4. Visualized development of product and process models. The design cases illustrate product-process associations in certain projects.

Model details

The first model (Fig. 5), the simplest one, is built from classes that represent the main parts of the gear

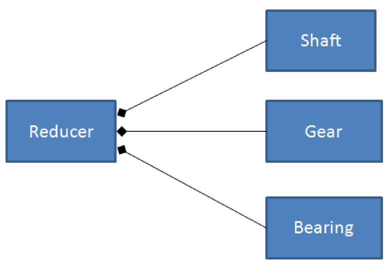

Fig. 5. First reducer model.

reducer: the *Gear*, the *Shaft*, the *Bearing* and also the *Reducer* class, which is a collection of other objects. The created objects contain data which is necessary to generate a geometric model of the suitable parts. However, this model does not include any information how these parts are connected or related with each other.

The second model (Fig. 6) contains more details of the reducer structure. This model does not only store the geometric parameters of the parts but also data concerning their mutual connections.

The third model (Fig. 7) is very similar to the previous one. The representation of the shaft is the only noticeable difference. The new model allows to describe the form of the shaft automatically. The development of the shaft model is presented in the further part of this work.

The first model of the shaft is presented in Fig. 8. It is the simplest model that can be created and it models the shaft as a list of shaft levels which are represented by the *ShaftLevel* class.

The second model of the shaft (Fig. 9) is based on the previous model and is extended by *ShaftLevelJoint* class. Instances of this class represent abstract points that can be placed at the beginning or at the end of the shaft, and also between each shaft level. These points can be used to create new shaft levels in the middle of the whole shaft.

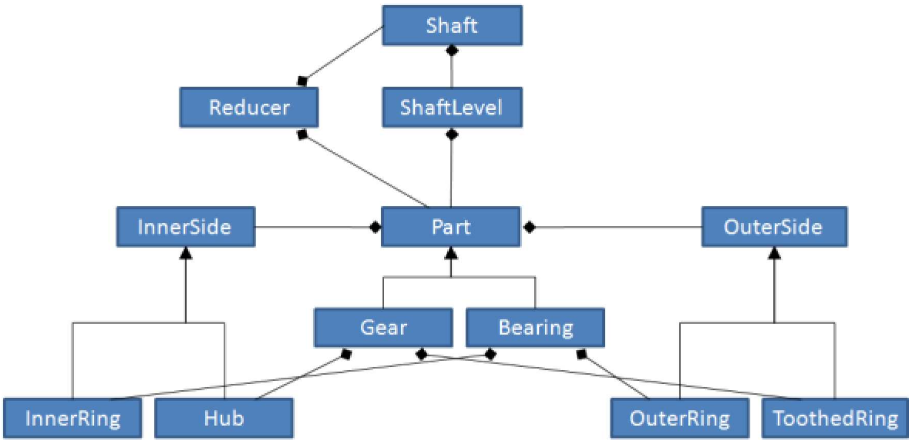

Fig. 6. Second reducer model.

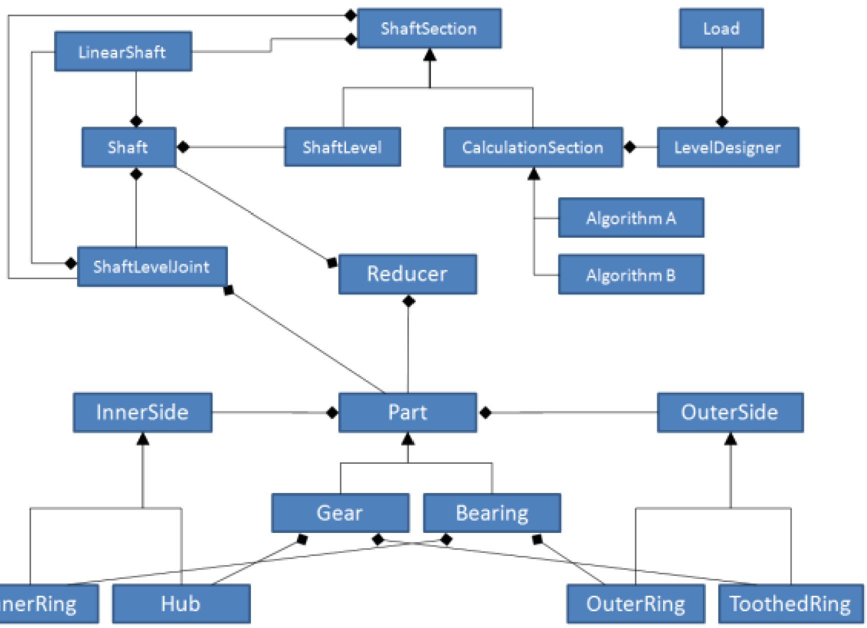

Fig. 7. Third reducer model.

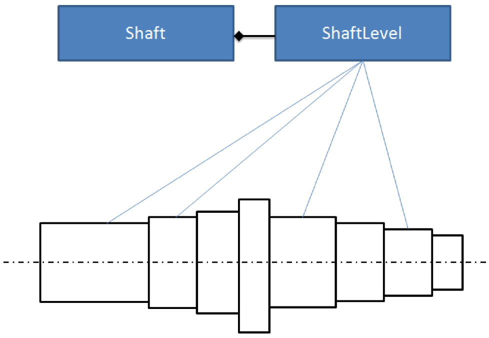

Fig. 8. Simplest shaft model.

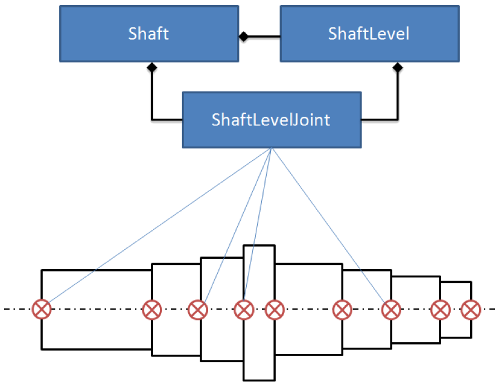

Fig. 9. Second shaft model.

They can also be applied to position other parts (gears, bearings) on the shaft. Unfortunately, using these points to place additional parts, it is necessary to check on which shaft level this part has to be placed. These points contain information about the previous and the next shaft levels and have to find which level has a smaller diameter.

The next model (Fig. 10) is again an extension of the previous one. In this model the new LinearShaft class was implemented. The whole shaft is divided into three main objects: a main shaft level with the largest diameter and two instances of the LinearShaft class on both sides of this shaft level. The object of the LinearShaft class contains a collection of ShaftLevel objects and assumes that the diameter of each next level is smaller than the previous one. This approach eliminates limitations of the previous model, because it is ensured that the next shaft level has a smaller diameter value.

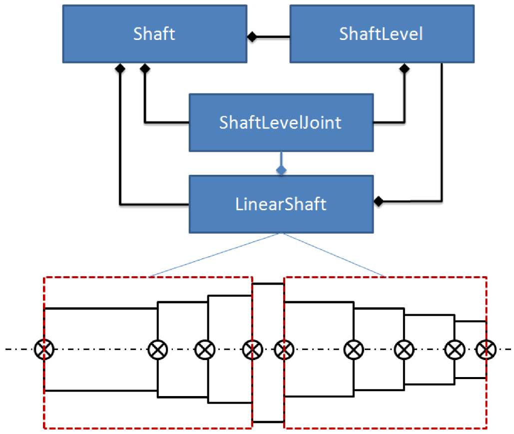

Fig. 10. Third shaft model.

The last model has implemented elements (Fig. 11), which allow for an automatic modeling of the shaft. The *CalculationSection* class is responsible for an automatic modeling process. Instances of this class can be created between two shaft levels defined by the user. Basing on the given load (Load class) and the selected algorithm it is possible to generate shaft levels.

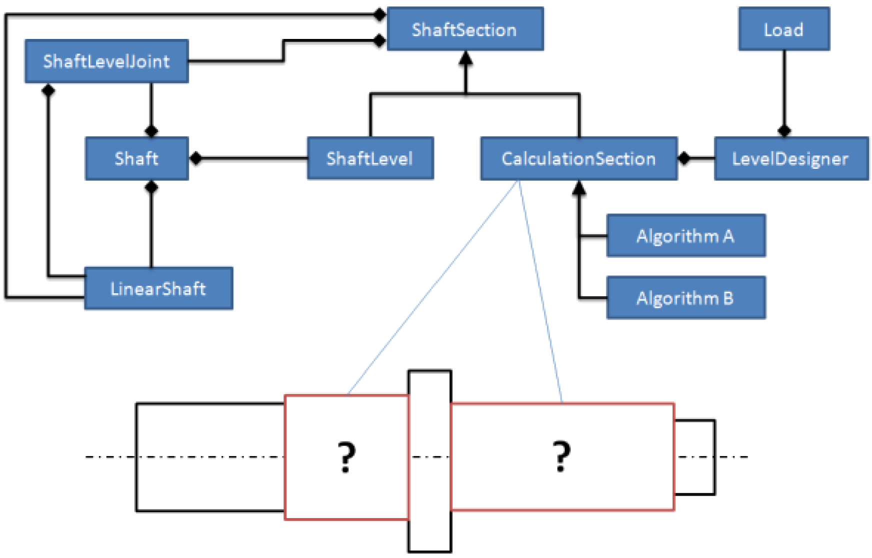

Fig. 11. Fourth shaft model.

Path forward

The approach presented in the previous chapters was originally meant to be used on more complex product/process like industrial design processes of automotive seats. The example of the speed reducer functions as a reference task for the basic concepts development and its presentation.

The authors assume that it is possible to build the geometric structure of a seat, to model its functions and its behavior. It is also assumed that it is possible to model these elements on different levels of their preciseness. For instance, the model can describe the general structure of the automotive seat and also different important details as separate object components integrated with the other elements of the product's model.

Conclusion

The most important feature of the proposed approach is its possibility to model only important parts of the models and to keep track of the development of their chronological paths.

The approach may be treated as a partial CPM/MOKA model or as network-integrated CPM/MOKA models.

The proposed solution allows to reduce the size of the stored and processed product/process information. Usually, CPM/MOKA models in case of complex objects are very large and not easy to deal with due to their permanent development.

The proposed solution seems to be very useful for models which change in time domain on their micro, local level and from time to time also on their macro level. The Finite Element Method (FEM) models of automotive seats have such characteristic features [11].

The number of the considered FEM models is especially huge in case of the world suppliers of automotive seats [11].

Of course, together with the base model other black-box models can be used or a human designer can function as a part of the model.

References

[1] Baysal M.M., *Functional and Behavioral Product Information Representation and Consistency Validation for Collaboration in Product Lifecycle Activities*, Mechanical and Aerospace Engineering – Dissertations, Paper 64, Syracuse University, 2012.

[2] ElMaraghy W., ElMaraghy H., Tomiyama T., Monostori L., *Complexity in engineering design and manufacturing*, CIRP Annals – Manufacturing Technology, 61, 793–814, 2012.

[3] Fenves S.J., *A Core Product Model for representing design information*, Internal Report NISTIR 6736, National Institute of Standards and Technology, Gaithersburg, MD, 2001.

[4] Fenves S., Sriram R., Subrahmanian E., Rachuri S., *Product information exchange: practices and standards*, Trans. ASME J. Comput. Inf. Sci. Eng., 5, 3, 238–24, 2005.

[5] Fenves S. et al., *CPM: A Core Model for Product Data*, Manufacturing Systems, Integration, Division, National Institute of Standards and Technology, Gaithersburg, USA Available at: http://www.mel.nist.gov/msidstaff/sriram. Accessed on: March 10th, 2007.

[6] Königs S.F., Beier B., Figge A., Stark R., *Traceability in Systems Engineering – Review of industrial practices, state-of-the-art technologies and new research solutions*, Advanced Engineering Informatics, 26, 924–940, 2012.

[7] Pokojski J., Woźnicki Ł., *KBE application development for selection-type design process* [in Polish], Mechanik, 7, 607–614, 2014.

[8] Gil M., Pokojski J., Skotnicki S., Szustakiewicz K., *Computer supported proces of Knowledge Based Engineering application development in engineering design* [in Polish], Institute of Machine Design Fundamentals, Warsaw University of Technology, 2011.

[9] Clarkson J., Eckert C. [Eds.], *Design Process Improvement. A review of current practice*, Springer – Verlag, London, 2005.

[10] Stokes M., *Managing Engineering Knowledge, MOKA – project*, Professional Engineering Publishing Limited, London, 2001.

[11] Pokojski J., Woźnicki Ł., Dębniak M., *Application Development for Process of Mesh Generation in FEM* [in Polish], Mechanik, 7, 615–622, 2014.