

SCALING OF TECHNICAL SYSTEMS USING AN OBJECT-BASED MODELLING APPROACH

Wolniak, Philipp (1); Sauthoff, Bastian (2); Lachmayer, Roland (1); Mozgova, Iryna (1)

1: Leibniz University Hannover, Institute of Product Development; 2: Baker Hughes, a GE company

ABSTRACT

Companies that operate and manufacture products in the technical area are exposed to increasingly challenging market situations. The developed products should be individualized to the customers' needs while offering high quality at an acceptable price.

The temporal and especially economic claims are constantly growing, forcing the companies to develop a given product that matches the cost-side as well as the technical requirements in a short period of time. Following an initial development, it is often necessary to provide further product variants regarding a modified geometry or performance. A time and cost efficient way is the scaling of the initially developed product.

Existing scaling methods focus on uniform geometry changes, not taking into account influences from non-uniform requirement or geometry alterations. Therefore, this article proposes an approach on how to model and assess the outcome of a scaled assembly, based on the connection of individual scalable components inside an object-based approach.

Keywords: Scaling, Object-based, Computational design methods, Knowledge management, Product modelling / models

Contact:

Wolniak, Philipp
Leibniz University Hannover
Institute of Product Development
Germany
wolniak@ipeg.uni-hannover.de

Cite this article: Wolniak, P., Sauthoff, B., Lachmayer, R., Mozgova, I. (2019) 'Scaling of Technical Systems Using an Object-Based Modelling Approach', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.166

1 INTRODUCTION

For a continuous production of oil and gas, the exploration and development of new springs is proceed globally every day. Within the oil and gas industry there are several service companies who provide all activities within this context to the several oil and gas producing companies. Due to the special environment in this industries and exclusiveness of technology, the development of new products, especially of tools for drilling the wells, is done by the service companies themselves.

The special case in the development of these tools is the recurring development of products within defined geometrical increments, analogously to the typical series development. The use of these tools in the drilling process of a well requires several different diameter sizes of the rotational drilling tools. The reason for this is the necessity of stabilizing the drill bore by inserting steel casings and cementing the outer annulus (Figure 1). Therefore, typical and mostly standardized diameter sizes exist, in which newly developed tools have to be delivered (Hossain and Al-Majed, 2015). The function of the tool itself as well as the overall design concept for the most part stays the same as in the initial development. Still, a high amount of time and cost resources are necessary. Especially the recurring conceptual phase has a great impact, regarding the exact dimensional as well as topological shaping through simulation-based analysis.

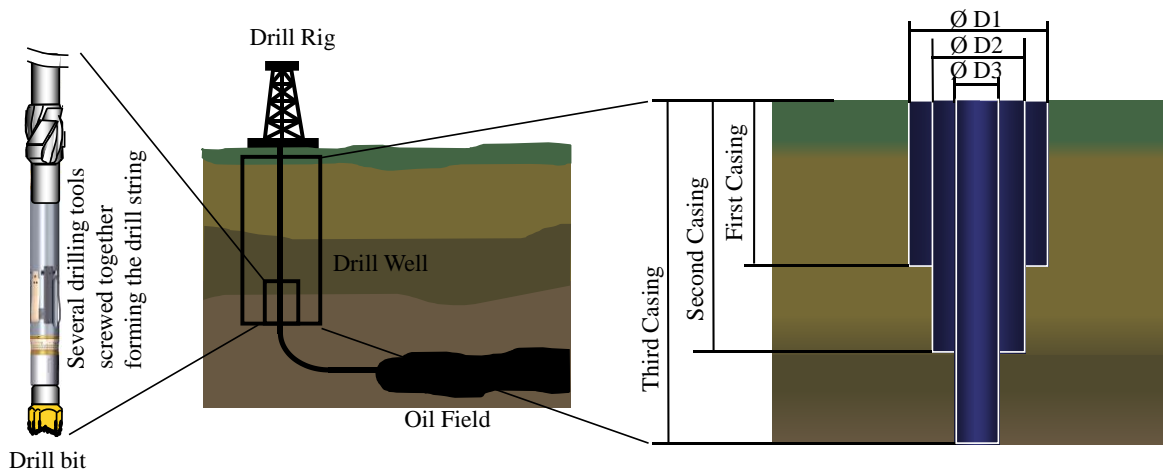


Figure 1. Principle of a well drill sequence and the resulting scaling necessity

A known approach of lowering the development efforts is the use of a series development with defined increments regarding the geometry as well as functional and performance parameters. The benefit is a wider range of product variants resulting in the possibility to respond to customer requests. A feasible proceeding in the series development begins with an initial development for a specified requirement according to the definition of a concept and an ongoing detailing. On this basis further variants are derived by using scaling equations and standard number rows to lower the development as wells as the later production cost through a standardized geometrical and performance disposition (Pahl *et al.*, 2007).

The scaling of a single component in a uniform as well as a partly non-uniform way is a topic described throughout the centuries in the literature. Still, there is no direct way described on how to scale a component geometrically fully non-uniform as well as on how to handle a non-uniform requirement change throughout the scaling increments (Wolniak *et al.*, 2018). Furthermore, the scaling of several independently changing components in an assembly is not handled in any manner. Hence, the focus in this article is on the connection between the scaling possibilities of a single component and the link of scaled components in an assembly.

Modern products often have a complex architecture, especially regarding the connection of several geometrical and functional parameters. The representation of the connections of these components as well as their functional and mechanical design are complicated to insert inside a single model. Thus, the approach of this article is to separate the domain knowledge on how to scale and design a single component from the process knowledge on how the scaled components interact inside an assembly.

While in the development and especially in the detailing phase of a product an explicit formulation of a model is desired, the conceptual phase is often performed with a more implicit approach (Weber and Werner, 2000). Many factors and parameters are not determined and have to be evaluated, often in

cumbersome iterations. Therefore this article proposes the use of the knowledge-based engineering (KBE) to insert domain knowledge on the design and the calculation of a single component for the specified requirements. This is done by inserting analytical equations and numerical solutions in form of case bases and linking these case bases to the KBE-system in order to obtain solutions, like mechanical strength assessments, for quick geometrical parameter changes. The overall goal is to establish a model-based environment with a defined library of single components containing the necessary knowledge in a KBE-system. The final step is the connection of these components in an assembly model to calculate and simulate the whole assembly.

2 STATE OF THE ART

The non-uniform change of the geometry of a component is problematic for deriving new variants by simple analytical calculation. While, with regard to the oil & gas industry, e.g. the diameter size of a drilling tool is doubled, the aim is to keep the length of the tool within a constant range. Especially the axial dimension represents a high cost factor regarding the material as well as the manufacturing effort. Another aspect is the occurrence of geometrically fixed components like electronic devices, which are inserted into the dimensionally changed geometry of the component under investigation.

In addition, the requirements regarding the mechanical, thermal or fluidic loads alter in different ways than the required diameter increments. Often requirements have static values due to technological progress, like a high temperature requirement, which has a certain value according to the possible highest temperature the mounted electronics can withstand. Other requirements change dynamically due to their affiliation to the dependence on customers' needs. An example is the mechanical loading condition, which changes according to the prevailing conditions due to the surroundings. Hard formations, narrow curves that have to be drilled or corrosive mediums contribute to a changing requirement according to the drilling location.

A description of the scaling literature and the usability in the presented approach is given in (Wolniak *et al.*, 2018). Regarding the proposed approach, the first step is to define the possibility of reducing the complexity of the system. This is performed using the support of a KBE-framework. In this context, the type of knowledge that is to be used has to be defined. (Schreiber *et al.*, 1993; Breuker and van de Velde, 1994) give a distinction of different kinds of knowledge, namely the domain knowledge, process or control knowledge and the inference knowledge. In this respect, the domain knowledge deals with problem-specific questions and can be seen as a declarative theory of the domain, like the actual analytical equations or numerical modelling of one specific component under investigation. By contrast, the process knowledge defines how to build up a system and implement the domain knowledge in the required positions. The inference knowledge is used in the problem-solving process and refers to the domain as well as the control knowledge to obtain appropriate solutions (Schreiber *et al.*, 1993).

To be able to separate these kinds of knowledge and to deal with the system complexity, this article proposes to divide the system of an assembly in single modules, containing solely the domain knowledge. Those modules represent logical segments of the assembly. The process knowledge then is completely separated and can be used to build up the system. The inference knowledge is used later on, while using the already existing system model.

Another important aspect of knowledge distinction is presented by Krause and Spur (1997) who define ways of knowledge representation, which are described as logic-oriented, constraint-oriented, rule-based and object-oriented. The selection of knowledge that can be used for component-specific problems and knowledge that can be used for the control of the system build up is performed by examining the possibilities to implement the desired information. While the first three representation forms are feasible for representing definite knowledge representations, the object-oriented representation has the flexibility to react to changing conditions. As the name indicates, the knowledge is represented by objects, containing information and methods which can interact with each other and therefore simulate complex systems (Bullinger *et al.*, 1989).

Following the combination of knowledge separation according to Schreiber *et al.* (1993) and the representation according to Krause and Spur (1997), the inevitable question is how to perform the system integration of these forms of knowledge. Therefore Ropohl (2009) presents three concepts of system environments, as illustrated in Figure 2. The first functional concept defines a system as an entity with states. The communication with the environment is possible based on inputs and outputs,

allowing an information flow. This concept is taken on in the presented approach by using capsuled objects, each representing a physical segment with the possibility to exchange geometrical or functional parameters with the environment or other objects.

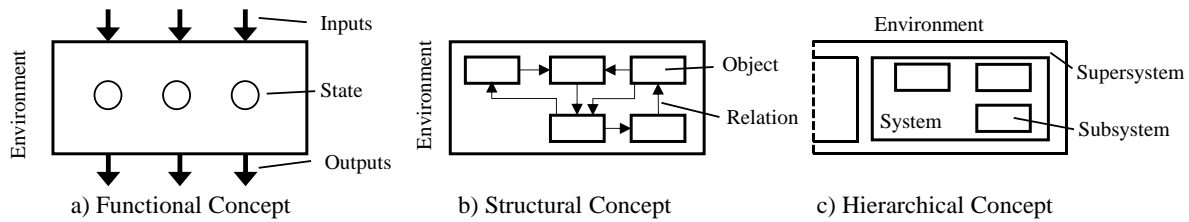


Figure 2. System concepts (Ropohl, 2009)

The second concept by Ropohl (2009) is the structural concept, which indicates the beforehand mentioned communication between every object. The third concept is the hierarchical concept, implying the combination of objects to set-up a subsystem with its own defined relations in a greater system. Combining these concepts and the knowledge representation forms makes it possible to obtain a formalizable way of building up a system containing modularized objects with domain knowledge for a comprehensive evaluation of an assembly.

An existing approach is presented by Stark and Königs (2014) deriving a template-based system development. By using an advanced SysML language, a variety of physical components inside an assembly is modelled. Especially dependencies and the connections between the components and their parameters are formalized and used. The main goal of this approach is to instantiate a formalized process of communication inside a company by integrating all development departments and assigning defined work orders at changes in the model.

Other works like (Muth, 1994) or (Bullinger *et al.*, 1989) are more specific on the integration of knowledge-based systems into CAD-systems and modelling assemblies with their parameterized dependencies. Conceptual changes throughout the problem formulation with largely changing geometrical sizes are not depicted.

La Rocca introduced the term “rule-based, object-oriented design” (La Rocca and van Tooren, 2010). Hereby, he uses so called high-order primitives for a conceptual variation of an aircraft design. His work focuses especially on the creation of variants and the preparation for the use in subsequent analysis tools.

A very broad and comprehensive description of the possibilities of the object-oriented modelling is given in (Hartmann, 2000), handling the mechanical illustration and calculation of civil engineering constructions and the planning of diverse assembly or building sites.

3 BASIC APPROACH

3.1 System- and parameter-level dependencies

Drilling tools are used in a modular way, so that they can be screwed together and therefore function as a long string (Hossain and Al-Majed, 2015). A schematic illustration of a drilling tool is presented in Figure 3 as a sectional view showing how the components inside this assembly are aligned. The top-level architecture of a drill string mainly results in a chain structure, whereby most components only have two direct geometrical neighbours.

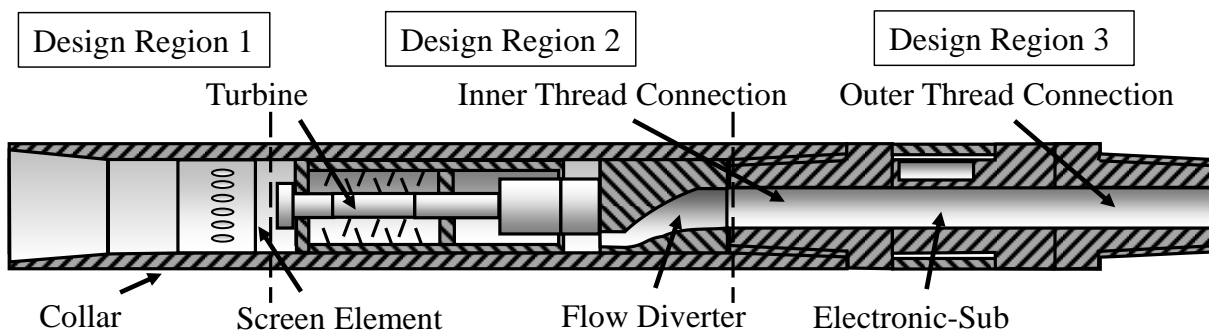


Figure 3. Example assembly architecture of a drilling tool

Nonetheless, dependencies throughout the whole structure exist. As a result, complex networks of dependencies and connections occur, leading to the fact that a strictly explicit formulation approach is not easy to obtain. Many dependencies have an impact on components in other areas of the product architecture, which are not manageable at once. Regarding a product with a multitude of components the variety of possible dependencies can become very high. Every component has mainly only direct neighbours in horizontal direction, as e.g. the inner thread connection is a part of the electronic-sub as well as it is also connected to the flow diverter and therefore sharing a common interface.

While this is the prevalent structure, there are also other connections which imply a more hierarchical approach. The flow diverter e.g. possesses a connection to the inner thread and to the turbine. Additionally, because the flow diverter is mounted inside the collar, a connection to the collar exist. As the dashed lines in Figure 3 imply, the approach is to divide the assembly in design areas distinguished by their function and therefore ensuring the before mentioned easier to handle chain structure through joint sub-assemblies (Sauthoff, 2017). The sub-assembly is achieved by combining the flow diverter and the collar and by treating this combination as one region. Combining all regions leads to the initial assembly, referred to as the system level. On the component level, the dependencies between the individual components still exist.

Figure 4 illustrates the difference between regarding every component and its individual dependencies on its own and the approach of combining design regions of the assembly as presented in Figure 3 by the dashed lines. It is noticeable, that the dependencies in the right part of the figure are not as clearly obtainable as in the left structure. Nonetheless, the combination of the function and the geometrical architecture of the product allows a better understanding and implementation of existing dependencies.

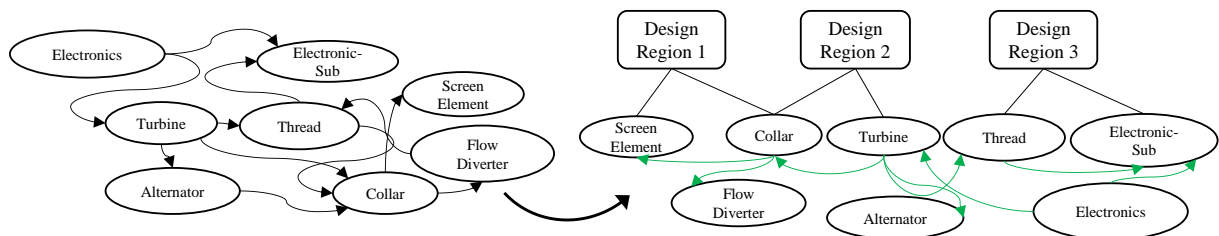


Figure 4. Dependencies in a simple chain and throughout the design regions

The idea of the proposed approach is to lower the complexity of the whole system by breaking it down to the component level and giving a methodical way of building the system of these fractions in a formalized way (Haberfellner and Becker, 2012). On the component level, it is necessary to formalize the domain knowledge, with regard to the design and calculation of this component. This is derived methodologically in Wolniak *et al.* (2018). On the system level and for the process knowledge an overlaying control mechanism is necessary ensuring a correct connection between the components.

The procedure of encapsulating the domain knowledge of a single component is inspired by the object-oriented programming (OOP) with individual classes having their own properties and methods (Booch, 2007). In the OOP, the distinction between a class and an object is made by implying actual values to the properties of the class, making this individual entity an object. Transferred to the presented model-based approach, every physical component with a defined set of parameters is represented by an object. The term property and parameter is used analogously in this context.

The communication of the objects in the OOP is realized by using a defined distinction of a recipient, a selector and an argument list (Muth, 1994). In the presented approach, every object's way of communicating inside the system is using specified in- and outputs. Thereby, the communication is analogous to passing information either from one object to another or from the system environment to the object (Haberfellner and Becker, 2012; Booch, 2007). The system description and consequently the parameter consistency is ensured by connecting all dependant objects. The type of information that is passed from in- to outputs is diverse, ranging from geometrical information, for the geometrical consistency of all components, to mechanical quantities like the mechanical loads or the stress-strain state.

To ensure the connection of the objects not only on a theoretical level but also in a way that can be performed computer-aided there are two logical levels. As mentioned before the connection logic on the system level ensures a general view on the assembly under investigation, allowing to distribute the general connections between the objects (Figure 4). While this is correct on a theoretical level it does not include the whole complexity and possible trade-offs between the objects under investigation.

Therefore a second level is necessary where according to the connections on the system level a connection network between the parameters of two objects is build (Figure 5, lower part).

These connection networks are not rigid but have to be flexible, due to varying requirements and non-uniformly changing geometries. Regarding the design process there are several possible scenarios in which order the whole system is designed and changed. These scenarios are affected by a high variety of decisions, ranging from management rulings to delivery bottlenecks and more. This quantity of possibilities is either handled by the option of a scenario selection or a manual decision, as shown in the further chapters. According to the choice made the dependencies throughout the system are adjusted to the special case, with regard to the components and the system.

3.2 Shared parameters and trade-off decisions

Another aspect is the handling of parameter trade-offs, which can arise from different components, affecting their own parameters as well as parameters of other components. Figure 5 illustrates this problem using the sample assembly shown in Figure 3.

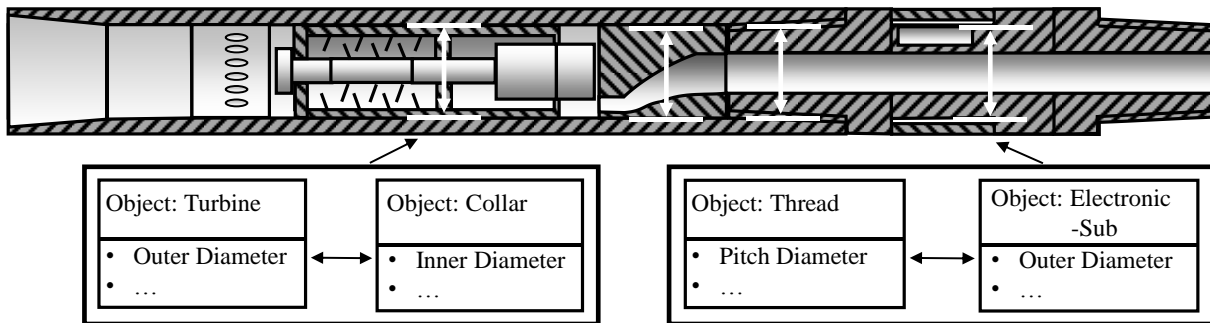


Figure 5. Parameter trade-offs of an exemplary drilling tool

In this special example, the investigated parameter describes the diameter of several components. The involved components are the turbine, the collar, the flow diverter, the thread and the electronic-sub building a multi-component dependency network. Although in every component this parameter is named and treated in a different way, it refers to the same diameter value. The question is, which component in the dependency network defines the value and therefore acts as the master to which the others have to adapt. As described before there are two possible ways of resolving this problem. The first solution is the automated model build up, where every trade-off is listed during this process. After the completion every trade-off between two components can be manually resolved, by choosing the master object. Following every decision the system is reconfigured and the next trade-off can be processed, leading to a resolved and trade-off free system. The disadvantage of this process is a high manual involvement and a high demand of design knowledge to correctly resolve the problem.

The second possible solution is the provision of predefined templates for various use cases. After choosing a template the integrated master assignment prevents any trade-offs and provides a solvable system.

4 MODEL IMPLEMENTATION

4.1 System level description

The implementation of the proposed approach is performed by using a combination of Matlab and the Toolbox Simulink. Figure 6 shows an example of a system consisting of three components (thread, electronic-sub, collar-probe). Thereby, every physical segment is represented by an object and displayed by the Simulink block representation.

These blocks possess inports and outputs, analogously to the inputs and outputs of an object. Between an output of a block and an inport of another block it is possible to draw a connection which is displayed by an arrow. Since the arrow is pointing only in one direction, from an output to an inport, the information flow is only possible along this line.

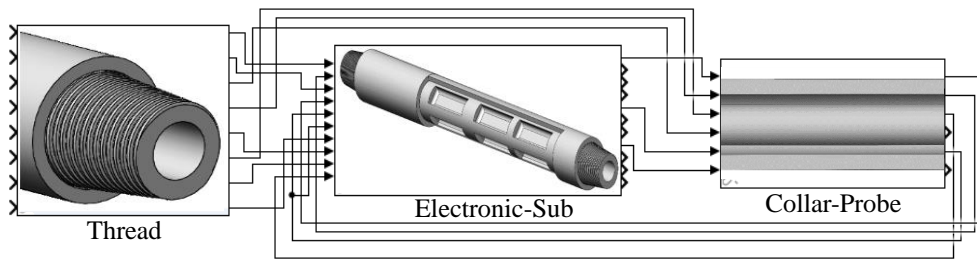


Figure 6. System-level description in the program implementation

As shown in Figure 6, it is possible to connect the objects in a forward as well as in a recursive way, allowing any kind of dependencies to be constituted. To prevent loops in the information flow it is necessary to monitor the connections on the parameter level and avert connections between parameters in two directions.

4.2 Object level description

While on the system level the connections are drawn to ensure the parameter consistency, the effects as well as the handling of these parameter information in the objects is not comprehensible in the representation shown in Figure 6. Therefore, Figure 7 gives an insight view into the object architecture and its methodical build up. In principle, the object is divided in two parts. The main block referred to as the electronic-sub is placed at the left side of the Figure 7. In the program implementation of Matlab/Simulink this block is programmed as a Matlab Level 2 S-Function (van Schijndel, 2014). This type of block allows the full control over the behavior and the design with respect to the inports and outports as well as the information flow through the use of Matlab scripting. The inports from the overlaying level of the system pass on the information to the main block and are connected to the proper ports of this block. In the main block the information are gathered and processed in a first step and then redistributed to the following solution blocks, placed at the right side of Figure 7.

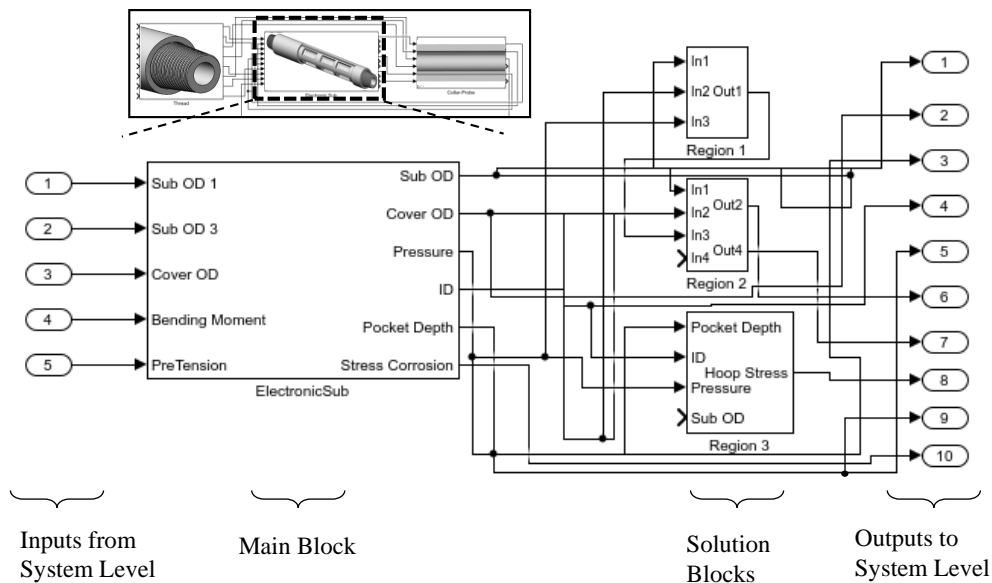


Figure 7. Object-level description in the program implementation for the electronic-sub

These solution blocks can have different constructions, ranging from simple underlying analytical calculations to large case bases containing a high amount of data distributed from numerical calculations. Moreover, these blocks can contain even a higher complexity according to the required result quality. Connecting these solution blocks ensures that the correct information is forwarded and therefore the correct solution is displayed.

Through the outputs on the right side of Figure 7, the previously generated solutions are distributed to the system level and can then be forwarded to other objects or displayed as values for further analysis. The number of solution blocks is varying, depending on the level of detail and thus, the resolution of geometry details and the underlying analytical equations as well as numerical solutions. A big

advantage of this fundamental structure of every object is the easy inclusion or exclusion of certain solutions regarding the desired outcome of the user. Furthermore, the subsequent addition of further geometry, calculation or knowledge detailing is easily applicable by adding the necessary type of solution block. This is done by inserting the information and knowledge in a generic way and connecting the solution block to the main block by using the correct ports and thus ensuring the correct information flow.

4.3 Parameter connection

To prevent the necessity of programmatically inserting and coding every possible connection between the objects, the idea is to use a matrix-based approach as presented in (Dörner and Bick, 1994). The matrix shown in Table 1 is analogous to the logical connections as depicted in Figure 4. Thus it represents the dependencies on the system level.

Table 1. Matrix-based approach for the dependency creation on the system level

	Turbine	Mud Screen	Thread	Collar	Electronic-sub	Flow diverter
Turbine	x	1	1	1	0	1
Mud Screen	0	x	0	0	0	0
Thread	0	0	x	1	1	0
Collar	0	1	0	x	0	0
Electronic-sub	0	0	0	1	x	0
Flow diverter	0	0	1	1	0	x

In this matrix, the rows as well as the columns contain one object per row and per column, arranged in an identical sequence, ensuring that every object is compared to every other object. The table is filled with either the number one or zero, representing the existence or non-existence of a connection, respectively. By declaring the object in the column as the dependent object, the matrix assigns directed dependencies in the network. The matrix is analyzed by an algorithm consisting of all implemented objects according to the user chosen components and returns the hierarchical structure of the selection in a simple vector form for further processing. Using this automated evaluation of the dependencies on the system level gives a simple way of controlling the outcoming dependency network.

Based on this dependency assignment the following step is the correct connection of the individual parameters of the objects. The approach is the same as in the system level, creating a matrix with a list of every involved parameter of one object. For every combination of two objects, one matrix is created, implementing the process and design knowledge of the dependency of these two components. Therefore every cell of the system level matrix contains an individual matrix with the parameter connections, which are activated if the dependency on the system level is selected.

5 CASE STUDY

In the following section two examples are presented to demonstrate the possibilities and advances of the proposed approach.

5.1 Example one: two-component calculation

The first example is a two-component composite of a thread connection and the previously presented electronic-sub in the context of new size development. Figure 8 shows the geometrical changes coming from the necessity of delivering a new drilling tool in the size 6 3/4" which is conceptually based on the previously developed 4 3/4" drilling tool.

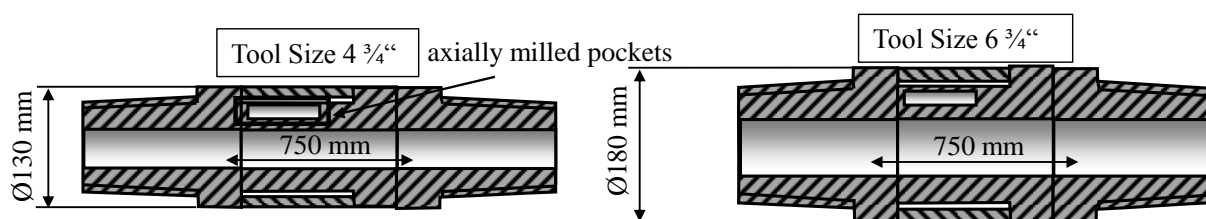


Figure 8. Scaled component sizes

As can be seen the outer diameter as well as the inner diameter of the tool are the main changing parameter. For other parameters like the overall length or parameters with functional determination the aim is to keep changes as small as possible or to use set values. This concludes in a geometrically inconsistent scaling and the necessity to assess the new results.

While this example is rather simple regarding the possible connections and trade-off decisions it gives a good insight in the actual possibilities of calculating and extracting formalized results. The system requirements as e.g. the temperature or geometrical limits can be inserted and are inherited as the defining boundary conditions for the objects. According to the inserted geometrical parameters and the dependency connections, the underlying analytical calculations and numerical approximations are performed. Hereafter, solutions as the stress-strain state in defined positions of the geometry of the electronic-sub as well as the thread are presented and are manually analyzable to assess the outcome. A simple cost calculation is included to help to compare different variants also from an economic point of view. If certain limits are exceeded, the inserted parameter values can be changed and another iteration can be calculated. After the necessary iteration loops, the final outcome is a defined thread and an electronic-sub which withstands the required loads and is in the geometrically prescribed range due to the required sizes and economic goals.

5.2 Example two: parameter change impact

The second example describes a fast way of overviewing parameter change impact. The assembly under investigation is the drilling tool as presented in Figure 3. According to the number and kind of electronics installed in the electronic-sub of the respective drilling tool, the necessary power distribution varies. The turbine, the clutch and the alternator as the main power supply unit have a crucial role in this consideration. Depending on the size of the turbine the power supply can be controlled. A possible way of adjusting the turbine is by changing either the diameter or the length by adding further turbine stages. Therefore, the necessary power distribution to the electronics and consequently the turbine dimensions can be seen as a driving aspect for a variety of components in the development of this exemplary drilling tool.

The presented case deals with the investigation of the outcome of a requirement change to a higher necessary power supply of the mounted electronics, also resulting in an either greater or longer turbine. Due to economic reasons, the preferable change is a greater diameter of the turbine, because of higher material costs when changing the length. A greater diameter of the turbine also implies a greater inner diameter of the collar, in which the turbine is mounted (Figure 9). Simultaneously, the thread connection is affected by the inner diameter of the collar, resulting in a weaker thread geometry. Since the thread connects the collar and the electronic sub, the diameter of the electronic-sub has to rise as well. The outer diameter of the electronic-sub impacts the rather thin collar, which is clamped above the electronic area of the electronic sub. Thereby, the overall thickness of the collar depends on the outer diameter of the electronic-sub. In this consideration the collar which is clamped above the electronic-sub represents the bottleneck and due to the shrinking thickness the strength becomes critical, especially regarding high outer pressure. This concatenation of dependant parameters shows how convoluted the impact of one parameter change may be on components in completely different regions of an assembly.

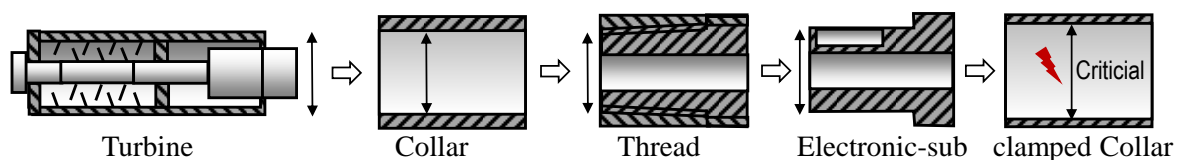


Figure 9. Concatenation of dependent parameters of the components

In addition, the impact of these changes can be calculated from a conceptual point of view and several iterations are performable in a short period of time, obtaining the optimal solution for this problem.

6 CONCLUSIONS AND OUTLOOK

Although scaling as a topic of research is thoroughly described in literature over the past decades, no exact method is given on the scaling of non-uniformly changing geometries and topology. This is why the approach of this article is to combine the knowledge-based engineering and the model-based

design in order to provide a development environment, where it is possible to perform non-uniform geometry changes within a parameter range and obtain a solution for further development.

The research has underlined the insights described in literature, that by dividing the domain knowledge and the process knowledge the complexity of the system, as well as invoked dependencies become better manageable. This is further used for a methodical way of building-up a configurable model of a large assembly. By subdividing the assembly into functional design regions the often complex dependencies are lowered to a simpler chain structure. Concluding, the presented model-based approach with the object-based representation of physical components gives a high flexibility of conceptual evaluation especially in the context of the assemblies they are built-in. This is contrary to the common scaling methods used in design with a rather procedural process and rigid structure.

Future work lays in the further derivation of individual objects and implementation into an object library. Using this object library several different assemblies are to be modelled to show the universal validity of the approach. Besides, the automated evaluation of the system results is of great significance as an effective way of supporting the design engineer. Using this evaluation the implementation of an optimization algorithm is possible to lower the development time and generate potentially optimized variants of individual components or the assembly as a whole.

ACKNOWLEDGMENTS

This research was supported by Baker Hughes, a GE Company.

REFERENCES

- Booch, G.O.-o.a.a.d.w.a. (2007), "Object-oriented analysis and design with applications", *The Addison-Wesley object technology series*, 3rd ed., Addison-Wesley, Upper Saddle River, NJ, Harlow.
- Breuker, J. and van de Velde, W. (1994), "Commonkads library for expertise modelling: Reusable problem solving components" / edited by J. Breuker and W. van de Velde, *Frontiers in artificial intelligence and applications*, Vol. 21, IOS Press, Amsterdam, Oxford.
- Bullinger, H.-J., Warschat, J. and Lay, K. (1989), *Künstliche Intelligenz in Konstruktion und Arbeitsplanung*, mit Verl Moderne Industrie, Landsberg/Lech.
- Dörner, D. and Bick, T. (1994), *Lohhausen: vom umgang mit unbestimmtheit und komplexität; [DFG-Projekt DO 2004 "Systemdenken", Lehrstuhl Psychologie II der Universität Bamberg 1981]*, Huber, Bern.
- Haberfellner, R. and Becker, M. (2012), "Systems engineering: methodik und praxis", Vol. 12., völlig neu bearb. u. erw. Auflage, Verl. Industrielle Organisation, Zürich.
- Hartmann, D., (Ed.) (2000), *Objektorientierte Modellierung in Planung und Konstruktion: Forschungsbericht*, Wiley-VCH, Weinheim.
- Hossain, M.E. and Al-Majed, A.A. (2015), *Fundamentals of Sustainable Ddrilling Engineering*, Scrivener Publishing, Beverly, MA.
- Krause, F.-L. and Spur, G. (1997), *Das virtuelle Produkt: Management der CAD-Technik*, Hanser, München.
- La Rocca, G. and van Tooren, M.J.L. (2010), "Knowledge-based engineering to support aircraft multidisciplinary design and optimization", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 224 No. 9, pp. 1041–1055.
- Muth, M. (1994), "Repräsentation von Konstruktionswissen unter Verwendung des objektorientierten Paradigmas", Diss, Schriftenreihe Produktionstechnik, Bd. 7, LKT Univ., Saarbrücken.
- Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Engineering design: A systematic approach*, 3. ed., Springer, London.
- Ropohl, G. (2009), *Allgemeine Technologie: Eine Systemtheorie der Technik*, Vol. 3, überarb. Aufl., Univ.-Verl. Karlsruhe, Karlsruhe.
- Sauthoff, B. (2017), "Generative parametrische Modellierung von Strukturkomponenten für die technische Vererbung", *Dissertation, Leibniz Universität Hannover; TEWISS - Technik und Wissen GmbH*, 2017.
- Schreiber, G., Wielinga, B. and Breuker, J. (1993), *KADS: A principled approach to knowledge-based system development* / edited by G. Schreiber, B. Wielinga, J. Breuker, Academic, London.
- Stark, R. and Königs, S.F. (Eds.) (2014), *Konzeption und Realisierung einer Methode zur templategestützten Systementwicklung, Berichte aus dem Produktionstechnischen Zentrum Berlin*, Fraunhofer Verl., Stuttgart.
- van Schijndel, A.W.M. (2014), "A review of the application of SimuLink S-functions to multi domain modelling and building simulation", *Journal of Building Performance Simulation*, Vol. 7 No. 3, pp. 165–178.
- Weber, C. and Werner, H. (2000), *Klassifizierung von CAx-Werkzeugen für die Produktentwicklung auf der Basis eines neuartigen Produkt- und Prozessmodells*.
- Wolniak, P., Sauthoff, B. and Lachmayer, R. (2018), *Scaling of Structural Components by Knowledge-Based Engineering Methods*, in May, 21-24, 2018, Croatia; The Design Society, Glasgow, UK, pp. 1757–1768.