www.cadanda.com Computer-Aided Design and Applications

Decreasing Engineering Time with Variable CAD Models: Parametric Approach to Process Optimization

Goran Kukec[1]

[1]S Tempera d.o.o., gkukec@tempera.hr

ABSTRACT

Evolution and level of product development in modern market demand more efficient approach to product design. A methodology to reduce design delivery time through the overlapping of sequential activities and implementation of engineering knowledge in means of variable CAD models was observed. Different techniques and approaches were observed for understanding of the problem. Custom-built variable model with GUI was compared to the conventional modeling technique on an example problem from the automotive industry. Comparative analysis of modeling techniques shows that there is a significant reduction in the design process. If this approach is used as a framework in the company design process it will definitely lead to downsizing the design time as well as it will optimize the whole engineering process.

Keywords: process optimization, variable CAD, parametric, FBD.

1. INTRODUCTION

Different customer demands and tailor-made products are nowadays one of the most important parts of product development and tasks of R&D department. To fulfill these demands in shortest time possible, and/or to give reasonable response with lowest cost possible, company should have the possibility to make variable products.

Feature based modeling and parametric approach basically gives the opportunity to every user to make variable models. But existing Computer Aided Design (CAD) software is limited to well-defined problem in narrow domain and possibly cannot fulfill all company needs or cannot be easily integrated as a sub-system.

Variable modeling is basically present in its basic sense in any parametric modeler. Feature based modeling offers functions as patterns, family tables, mirroring and etc. Some of today software offers ready-made user-friendly oriented features, as i.e. variable pattern in Solidworks, shown on Fig. 1. Of course, no matter the function is added as the ready-made feature or not, any of these features are implemented to the model by the user. This means that the user needs an appropriate amount of knowledge, skills and/or experience to make this procedure happen. Mathematically speaking, in a parametric model, each entity, such as a Boolean primitive, has parameters associated with it. These parameters control various geometric properties of this entity and their locations, such as the dimension of a body, or the diameter and a position of a hole. They also control the non-geometrical parameters of the model, such as the name of the body and its modeler. These parameters can be changed by the operator as necessary to create the desired part or can be controlled automatically (i.e. with an equation).

Additionally, history based modeling implements relations between features based on the modeling process timeline. This, again, requires not only the knowledge and skills but also the understanding of the model functionality, co-relations and the timeline.

Parametric modelers that use a history-based method also preserve a record of how the model was built. When the operator changes parameters and regenerates the model, the program repeats the operations from the history, using the new parameters, to create the new model. Optimization of the model timeline can often make a huge difference between end-models (in the sense of the space they "weigh"). But, this should be observed separately.

Variable modeling implements various rules, co-relations, relations, boundaries and equations in means of assigning the values to the parameters which results in built-in knowledge in the variable model itself. There are many uses for this type of

Computer-Aided Design & Applications, 12(5), 2015, 651–661, http://dx.doi.org/10.1080/16864360.2015.1014744

Taylor & Francis Taylor & Francis Group

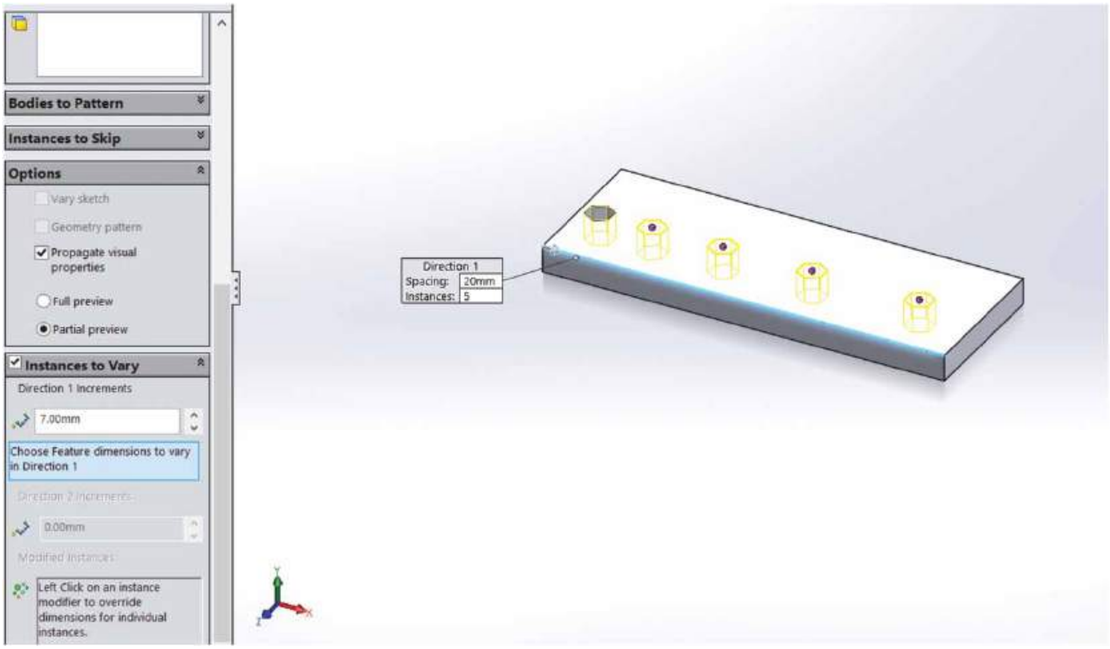

Fig. 1: Variable pattern, Solidworks 2014.

modeling. Not only designers can test variants of their models to determine which is the "best" solution by simply adjusting the variable parameters and regenerating the part by a click of a mouse, the process of the modeling is getting much closer to the wider audience and the possibility to make an error should be smaller.

Remaining question is can it actually lead to the modeling process optimization and is it worth it?

2. MAIN BODY

Today designers are forced to reduce the design time so production could proceed sooner and overall project time would be reduced. Not to mention that managers use various methods to estimate designers a priori their employment.

To reduce the modeling time, the modeling process itself must be observed. Easiest way to reduce the whole process is to find a way to loose steps or the whole branches in the process. While conventional modeling consists of a whole tree of branches and demands from the user, variable modeling consists only from inputs. As it is logical to conclude that inputs will take less time than Feature Based Design (FBD), due the prior "Knowledge" step is also not needed, the hypothesis was that variable modeling needs less time than conventional.

2.1. Understanding the Conventional Modeling Process and Time Optimization

As described earlier, the most demanding step in conventional modeling process (Fig. 2) is the "Knowledge" step. This step includes implementing the marginal values, rules, relations and co-relations as well as different types of information to the model (parameters). Often, this takes time. In some companies, this step is partially shortened by smartly created templates. Good example of simple waste of time is the "Author's name" parameter. This parameter can be solved with templates (as well as a few other good solutions) and there isn't actually a need to fill in all your data every time you create a model. Never the less, in some companies, people still fill in their names every time they create the model. The point is, if you find a pattern of "standard" values for some parameters, and of course, assign those values to the corresponding parameters, there is a possibility you will spare some time and also minimize the possibility to make an error.

While there is no universal formalization of the design process to develop a framework based on concurrent engineering principles, [7], which could be used to evaluate co-relations in design process, this is still a sustainable reduction in design delivery time and it can be used as a company set of rules.

Next step is the FBD itself. As this step includes the understanding of the model itself and the whole know-how concept behind it, this step must be observed for each model separately and often depends on the designer itself. Nevertheless, there is always a chance to shorten your final time by assigning a few options and values to the parameters.

2.2. Variable Modeling Process and Time Optimization

Main advantages of variable modeling process are the pre-implemented marginal values, rules, relations and

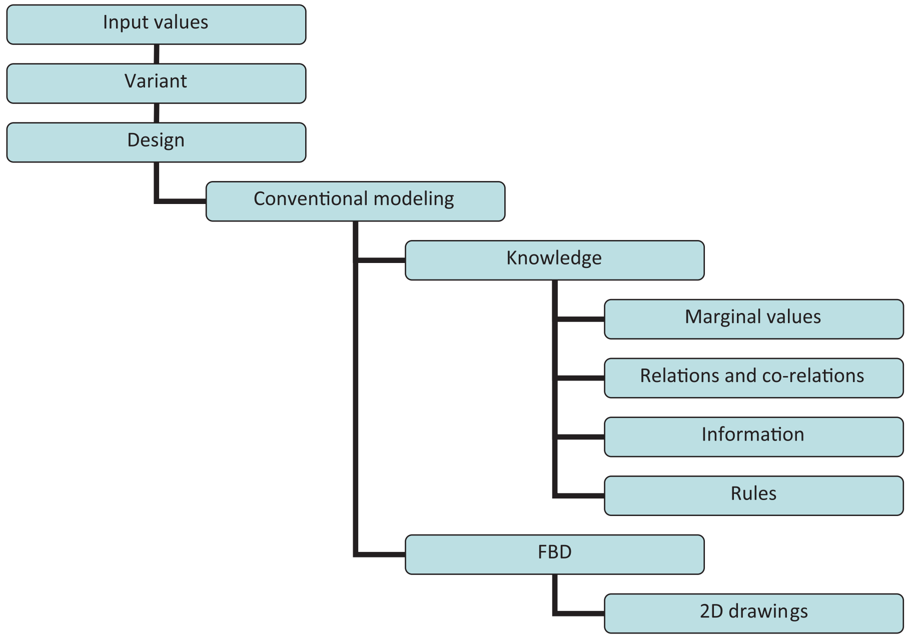

Fig. 2: Conventional modeling process.

co-relations as well as different types of information to the model (parameters). In one word – knowledge. This spares time.

On the other hand, this procedure can be controlled at some point by company or design rules/procedures. This obviously means that the knowledge is implemented to the model as a prior – boundary condition. It is also very important to mention the potential risk of knowledge sharing and misunderstanding between team members (in this case – engineers). Some authors describe this phenomena as "not speaking the same language effect", [3]. If we gather co-relations, restrictions, rules and engineering knowledge in general, of symbolic and topological information to the variable model as a core – database functional action model then it is possible to make clear, functional and fast sub-system integrated to the existing parent system.

It is very important to make clear and understand the procedure of variable model building, as it is completely different than conventional modeling technique. In conventional modeling, operator assigns parameter values by adding different features to the model step-by-step, while in variable modeling operator assigns all values before any building starts. For the computer solver this approach implies that all equations must be solved after all the values are set, which will result in time usage for model building. There is a possibility to run the solver before all the values are set but in this case there could be other problems with sorting out co-related parameters which will result to solver crash and definitely results in larger programming complexity.

There are a few commercial solvers integrated into the parent system, [2, 6]. Practice has shown that all solvers are efficient for small problems. But, there can be big differences in performance (speed and memory usage) in solving large problems. One of the most commonly used methods is the iterative method. User defines variables, constraints and the goals. Solver uses iterative method to pick-up the data for different scenarios inside given constraints. In the end, the system compares the picked-up data from the scenarios and compares them. Finally, you can choose the best solution.

Depending on the complexity of the study and the solver itself, designer can spare time not only he would "normally" spend on the design but also he has a variety of data for different scenarios. This data can be used for various plots, reports and etc.

Complex assemblies with many features are numerically difficult and also certain expertise may be required to overcome them. It is difficult to predict when this issues may appear. In any case, one of the most common design problem is the design optimization, which leads us to the iterative design model itself.

2.3. Parametric Approach and Iterative Design Model

We could say that every parameter has its own definition as a type and the assigned value. Parameters can be divided to internal parameters (needed for the model – i.e. dimension) or user parameters (needed for the user – i.e. price). Parameter type can be string, integer or Boolean.

User definition of a parameter is very important due the different types of parameters could have problems with relations and mathematical equations (i.e. when using arithmetic operators).

Value can be assigned to the parameter by the user or it can be assigned automatically (i.ee with an equation or a function). Again, parameters can be divided to constants and to the variables. This part has to do with the program itself, as well as earlier mentioned knowledge/rules implemented to the model. Relations or sub-programs control variables, depending on the co-relation complexity.

Knowledge implementation by assigning the value to the parameter can be done as a constant value or as a function. This can be explained with a simple IF THEN ELSE statement (please look at the Fig. 3). Imagine there is a “normal” parameter – d0. When user inputs the value of parameter d0, system automatically assigns a value to parameter d1, depending on the value of the parameter d0. With different expressions, different scenarios can be solved. This implies that a sub-program can be made to control the assignment of a value to the parameter.

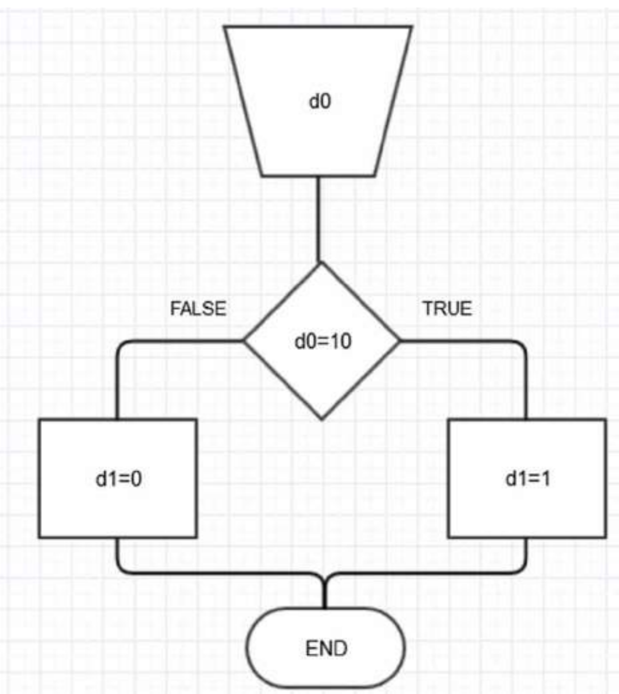

Fig. 3: Parameter value assignment by IF THEN ELSE function.

While earlier stated problems in this paper are mainly concentrated on boundaries, described as marginal values, company rules and etc., in section 2.2, optimal design idea was mentioned.

Finding the optimal solution is not easy. There are many methods to do it. But, if we look at the explanation of assigning the value to the parameter as a function, we could analyze it more thoroughly. So, if the system assigns the value by an expression (in earlier mentioned case by IF THEN ELSE statement) why couldn’t the system do this as an iteration? In theory - yes and it is very easy. For understanding the idea, a simple iteration process is explained in Fig. 4: value d0 must be assigned until required condition is fulfilled.

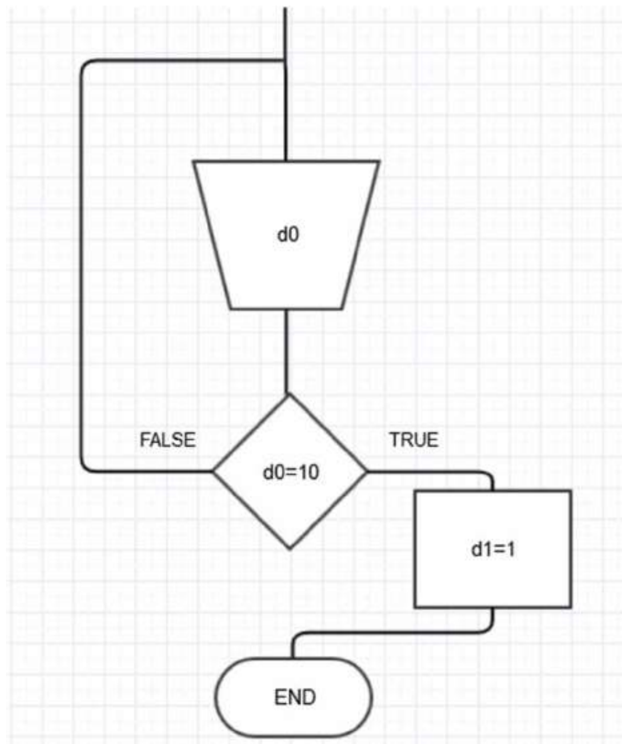

Fig. 4: Optimization model by iteration function.

In practice, when having more than one condition and a lot of parameters it is likely people will have trouble understanding the model behavior and setting the target values. With a small human mistake, program can offer non sense solutions. Not to mention the program crashes.

Solution model can be made on this basis (Fig. 5). Total optimization time will depend on the number of iterations (and steps between them). Each iteration consists of three stages:

- Definition stage
- Design stage
- Evaluation stage

In the first stage – the Definition stage, operator must implement knowledge into the model. This means setting the marginal values and assigning the input values. In variable modeling, it is the most important to understand this stage and to set the values correctly. Normally, this step is kindly to take

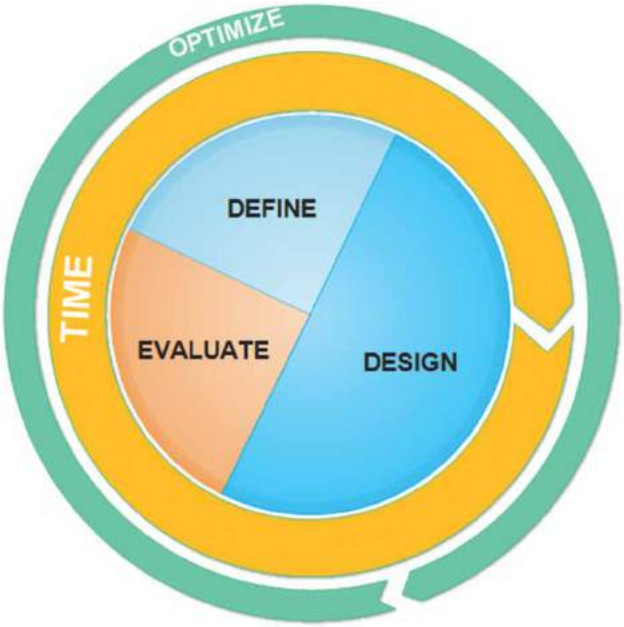

Fig. 5: Iterative optimization design model.

more time than it would take in conventional modeling process.

Design stage is the stage in which variable modeling should speed the design optimization process. Human interactions must be minimized in this process. Depending on the model complexity, number and complexity of functions, relations and co-relations, as well as the software language optimization and hardware which is used this step will make differences in the timetable.

Evaluation stage can be both taken by the human and the computer. In any case, understanding of the results and criticism is vital in this stage.

It is hard to make a general statement will this kind of design model lead to any timetable improvements. In any case, the possibilities are different and even though there is a possibility that the whole process will make just an incremental optimization, it will make a difference.

2.4. Custom Built Variable Models

While commercial software concentrates on wider audience and usage possibility, tailor-made solutions can be made for target-purposes. In a customization situation, the application is by definition unique. Therefore it is possible to implement as much as knowledge possible. If we observe the earlier mentioned rules implementation, the possibilities are huge (Fig. 6). Not only the manager can ensure that designer is following the company rules but he will take less time spending on the approval of the documents (as they will for sure look the same and contain all the data needed no matter who is the designer). Also, the production/technology department will have less problems.

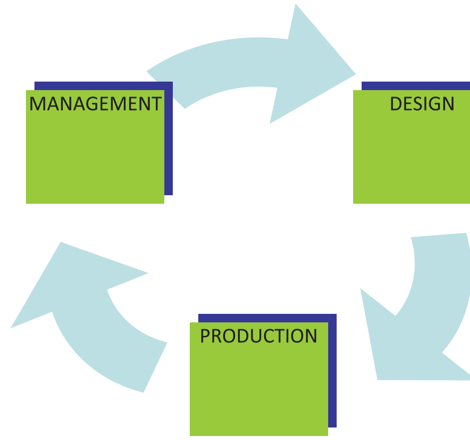

Fig. 6: Variable optimization per user relations.

Of course, some problems will be first-time experiences for all involved, and therefore may be time-consuming to resolve. Potential risk in different approach, in general or just to parameter definition and/or theirs understanding could be connected with engineering experience and practice, [4]. It is inevitable that adaptation to the new system will be time consuming. Another issue involves the need for extensive programming, which could result in large costs for the company.

With tailor-made products, there is always a problem they are vulnerable to paralysis of operations if the customizer-programmer makes himself scarce at a crucial time. In this case, when we are talking about a sub-program, even bigger problem is if the big-corporate vendor decides to shift strategy and makes changes to the main software.

2.5. Case Study

Products from automotive industry were observed. Aluminum sub-assemblies made from main aluminum panel are connected together with forged parts and screw bolts (Fig. 7). This makes the base of a truck superstructure. Mainly, these bases are used inside the company for production of fire-fighting trucks. In this paper, scope of observation was the aluminum panel sub-assembly design for fire-fighting trucks on commercial-type chassis. Different variants of observed sub-assembly can differ in size, quantity and even number and part variants (Fig. 8).

Using the assembly variant analysis, which was made as a questionnaire with the advanced users, analogy between models was found. Information was used to define certain parameters as constants.

As an example, holes on the aluminum panels can be observed. There are two types of holes. First group are technical holes – which are used for assemblage

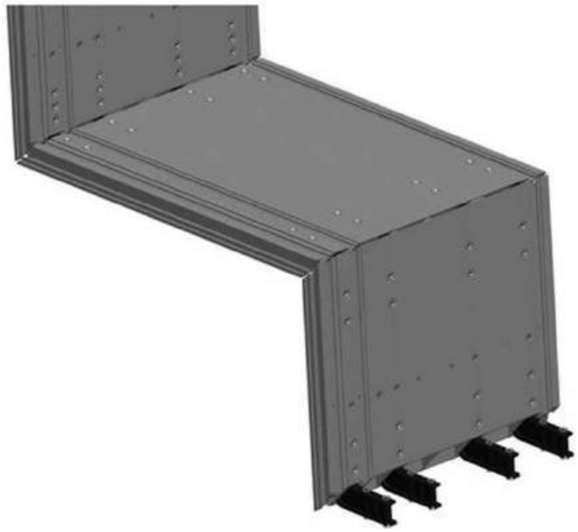

Fig. 7: Aluminum panel screw-type connection example.

and mounting of additional components. Other group are the technological holes – which are used only during the production process (i.e. for aluminum panel conservation). Holes, in general, can differ in quantity, size and absolute position (position of the holes on the aluminum panel) while the depth of a hole is always the same. But, technical holes are always positioned with the same step (one hole relatively to the other), both longitudinally and transverse to the aluminum panel. (Number of holes are always the same in each row and column). The remaining variables of technical holes are the total number of holes (in a one row) and the diameter of a hole. If technological holes are used (they can be used but they are not always used), then they are used always in the same quantity and they are always positioned transverse by the certain rule (which can be explained as a mathematical function). The few remaining variables of the technological holes are the diameter of the hole and the position of the holes relatively to the longitudinal aluminum panel axis.

The table below shows the total number of parameters used to define technical holes on the observed model and the number of values (variables) which must be assigned inside the variable model.

Tab. 1 shows that the total number of parameters which must be assigned to the model by the user during the conventional modeling is 7, while with variable modeling is only 2. Also, we must consider that during the conventional modeling, user must have certain skills and/or must follow certain rules, which will definitely take additional time during the modeling process itself. On the other hand, in variable modeling, all possible rules and knowledge is implemented inside the modeller itself. So, we can assume it will take less time to fill in the data needed for the user with the same skills or user with fewer skills could

No.	PARAMETER: HOLE	TYPE
1	DIAMETER	**VARIABLE**
2	POSITION, LONGITUDINALLY	CONSTANT
3	POSITION, TRANSVERSE	CONSTANT
4	DEPTH	CONSTANT
5	TOTAL NUMBER	**VARIABLE**
6	STEP BETWEEN HOLES, LONGITUDINALLY	CONSTANT
7	STEP BETWEEN HOLES, TRANSVERSE	CONSTANT

Tab. 1: Technical holes, parameter types.

do it. Same analogy can be used on the technological holes parameter analysis also, as well as the analysis of all other features. Of course, parameter names, types and number will be different.

Core parametric model was made in feature based parametric CAD modeler. Parameters are internal parameters (needed for the model – i.e. dimension) or user parameters (needed for the user – i.e. name), so end functionality of the model remains the same as with the conventional modeling process. Parameters are divided in variables and constants. Relations or sub-programs control variables, depending on the co-relation complexity.

In this case, the whole Graphical User Interface (GUI) was made (Fig. 9), which gives the possibility to adapt to specific needs of any company/department as well as the fact that every new user can clearly understand the model, its parameters, values, prices and in the same time it takes away the ability from the user to assign meaningless values to the model without prior expert knowledge and experience. With this approach, user can make variable models with basic license without need for additional modules, which would conventionally be used for variable modeling. Drawings are re-generated automatically according the changes made to the model. Remaining problem are the dimensions. This problem was not solved completely due to the different parts, their shape and position and it is left to the user to open the auto-generated drawing and to put the missing dimensions manually, the same it would be done on every other drawing.

2.6. Testing and Comparative Analysis of Custom Built Variable Model with GUI and Conventional Modeling Approach

2.6.1. Understanding the conventional modeling approach

With conventional modeling operator usually starts by modeling the first panel. First panel is usually the panel above the rear vehicle axle, as for this panel it is usually most easy to set it in assembly

Fig. 8: Observed aluminum sub-assembly, two variants.

space (engineer must set the global coordinate values and define the position of this panel inside the final vehicle assembly). After that, other panels can be modeled and connected by adequate mates into the assembly. This takes the most of the time. Also, the operator must fill in the values of non-geometrical parameters. While doing this, operator must define some values after reading the data from the previous models. (I.e. article numbers, which are given by ascending sequence.) This step is also very important and it is interesting that this step is the step in which most-likely errors happen. In the end, operator must set a number of section and detail views, as well as set the adequate scale to start making the drawings.

2.6.2. Custom built variable model with GUI

Operator starts modeling by starting the GUI, [7]. After the choice of the shape variant, default model is being loaded in the background, with all known parameters. Remaining parameters without set values must be filled by the operator. Mainly, those are dimension and yes/no parameters. System also demands the final assembly position in relation to global coordinate system – which is a counterpart step to the first step in conventional modeling. All the non-geometrical parameters which can be assigned by the non-human, are assigned automatically to save time (i.e. article numbers are set by the software by ascending sequence). In the end, all the values are set and the model is re-generated. Default section views, scale, detail views and the dimensions are made in drawings.

2.6.3. Testing and results

Testing of the variable model reliability itself was made prior in comparison to the existing technical documentation and possible values as well as by method of refutation by combining marginal and not-allowed values/variants. This testing method helped to determine bugs, faulty equations, missing relations and not well-defined knowledge integration.

Time reduction and process optimization tests were made by giving the variable model with GUI to

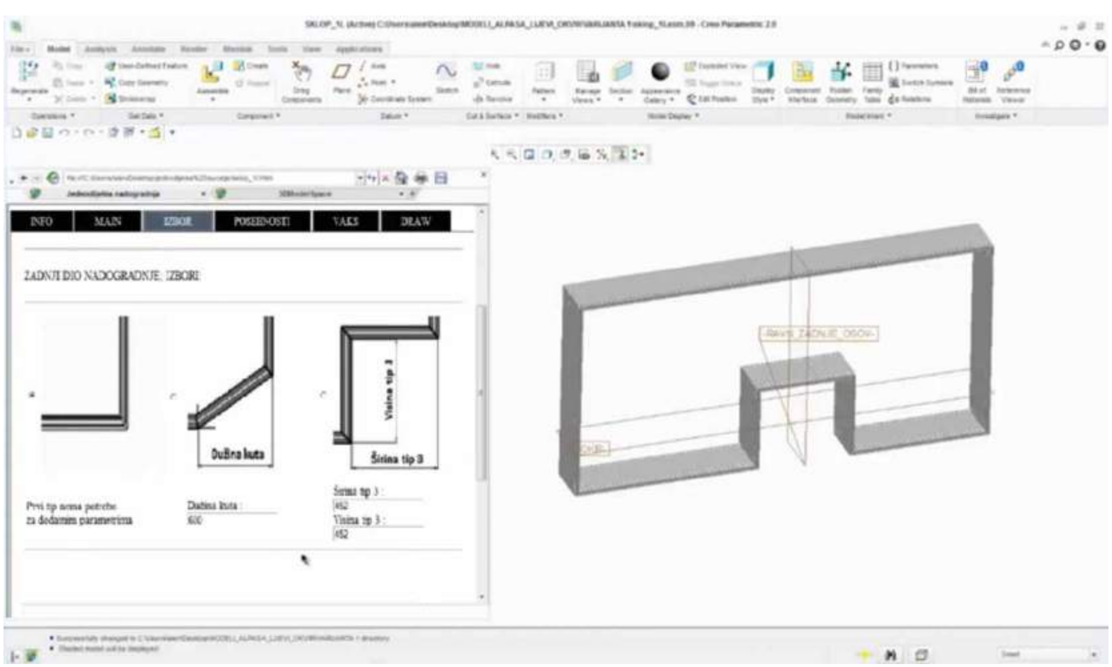

Fig. 9: Custom built variable model with GUI implemented inside PTC CREO.

the R&D engineers in testing period of two months. Data was observed and processed.

Testing of time needed was made for three different shape variants with different dimensions and different number of parts. Totally 5 users (mechanical engineers) were involved in the study. Data was collected.

Three different shape variant possibilities which were observed in this case are shortly described, along with their core variable and constant parameters in the Tab. 2.

Computer processing time for building the model was not involved in the study, as there is a large difference in model building methodology between conventional modeling technique and variable modeling technique, explained in section 2.2. Therefore, it is very hard to collect and to compare the data between.

Working hour analysis showed that average engineer needs roughly 28 hours (3 ½ work days) to make functional 3D model and 2D technical drawings for each part and sub-assembly. Main idea was to reduce this time (cost).

Significant time savings can be observed by each shape variant and by each user even on the 1st use. While there is a rising tendency in time optimization accordingly to number of use per each user and

Shape variants examples	Picture
Variant No.1 – stepped back – Only 90° connection elements are used – Aluminum panel ends must be cut at 45° – Adequate holes must be drilled on each panel for screw-type connections (differences in hole positions, size and number) – Each panel can differ in length – Assembly could be central positioned above the rear vehicle axle – Holes for external equipment – Symmetric left and right side?	
Variant No.2 – flat back – Only 90° connection elements are used – Aluminum panel ends must be cut at 45° – Adequate holes must be drilled on each panel for screw-type connections (differences in hole positions, size and number) – Each panel can differ in length – Assembly is not central positioned above the rear vehicle axle – Holes for external equipment – Symmetric left and right side?	
Variant No.3 – angled back – Mostly 90° connection elements are used – Aluminum panel ends must be cut at adequate angle – Angle section can be standard (fixed by connection element at default angle) or can be deviated (fixed by welding the aluminum panels ate user set angle) – Adequate holes must be drilled on each panel for screw-type connections (differences in hole positions, size and number) – Each panel can differ in length – Assembly is not central positioned above the rear vehicle axle – Holes for external equipment – Symmetric left and right side?	

Tab. 2: Shape variants examples.

	Time needed (hours)					
Variant 1	Conventional modeling	Variable model, 1st use	Variable model, 2nd use	Variable model, 3rd use	Maximum time savings	Maximum time savings (%)
User1	25,00	11,00	10,00	7,00	18,00	72,00%
User2	21,00	12,00	10,00	9,00	12,00	57,14%
User3	22,00	12,00	9,00	8,00	14,00	63,64%
User4	22,00	10,00	10,00	8,00	14,00	63,64%
User5	20,00	11,00	10,00	9,00	11,00	55,00%
Average 1	22,00	11,20	9,80	8,20	13,80	62,73%
Variant 2	Conventional modeling	Variable model, 1st use	Variable model, 2nd use	Variable model, 3rd use	Maximum time savings	Maximum time savings (%)
User1	32,00	11,00	10,00	8,00	24,00	75,00%
User2	30,00	11,00	10,00	8,00	22,00	73,33%
User3	32,00	10,00	9,00	8,00	24,00	75,00%
User4	28,00	10,00	10,00	8,00	20,00	71,43%
User5	30,00	11,00	10,00	9,00	21,00	70,00%
Average 2	30,40	10,60	9,80	8,20	22,20	73,03%
Variant 3	Conventional modeling	Variable model, 1st use	Variable model, 2nd use	Variable model, 3rd use	Maximum time savings	Maximum time savings (%)
User1	30,00	10,00	8,00	8,00	22,00	73,33%
User2	32,00	11,00	9,00	8,00	24,00	75,00%
User3	32,00	11,00	9,00	7,00	25,00	78,13%
User4	30,00	10,00	9,00	8,00	22,00	73,33%
User5	28,00	10,00	10,00	8,00	20,00	71,43%
Average 3	30,40	10,40	9,00	7,80	22,60	74,34%
Total average	**27,60**	**10,73**	**9,53**	**8,07**	**19,53**	**70,77%**

Tab. 3: Modeling time comparison.

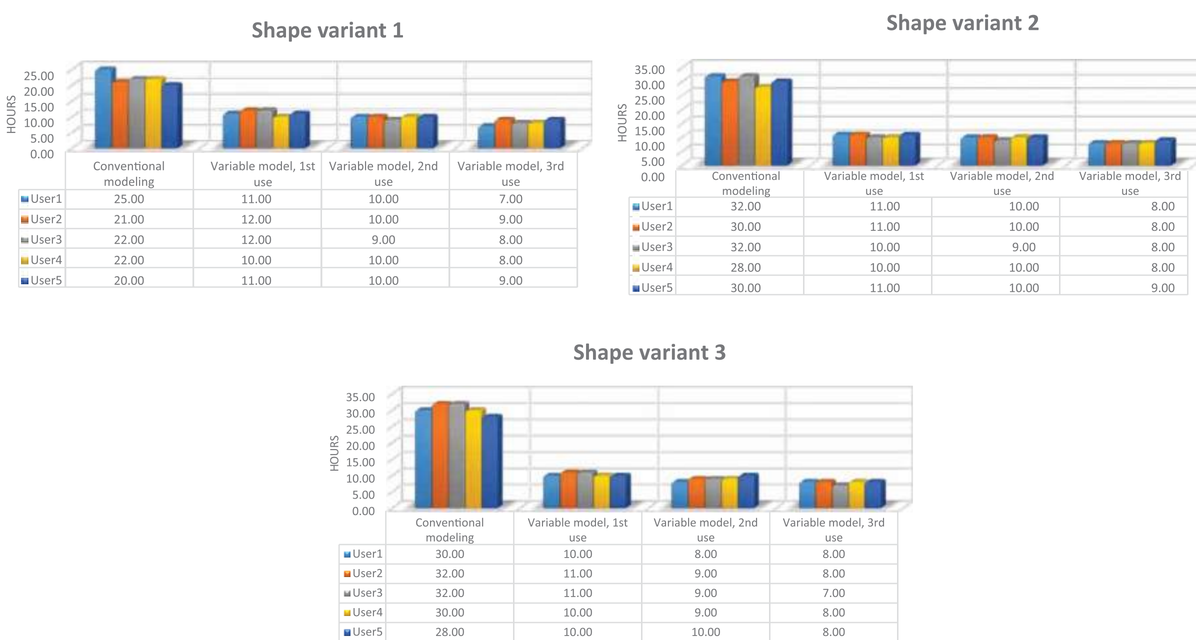

Fig. 10: Comparison of time savings between conventional and variable modeling technique using different shape variants, users and numbers of use per each user.

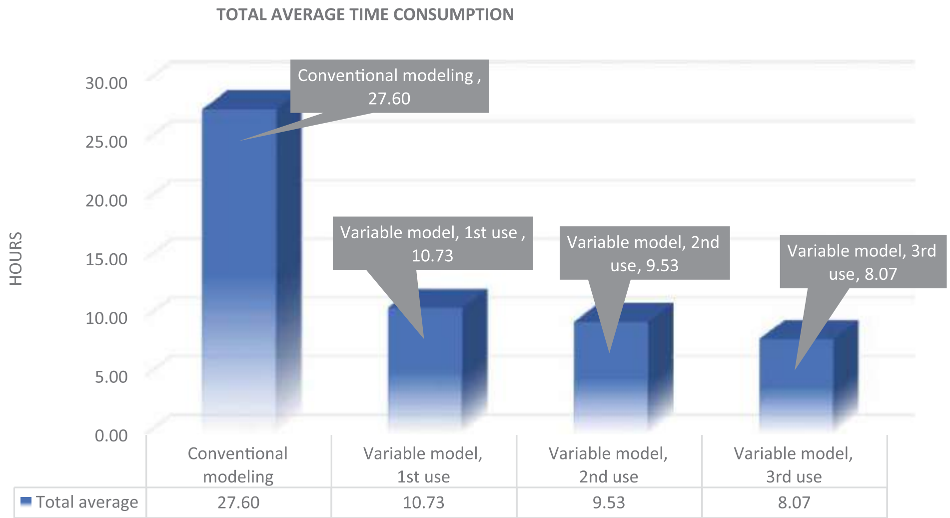

Fig. 11: Total average time consumption chart.

per each variant, for further conclusions, additional testing should be made.

Total average time (Fig. 11) was reduced from 27,60 working hours to 8,07 working hours, which is average total savings of 19,53 working hours or 70,77%. Maximum time savings were in 3rd use of each shape variant per each user. By comparing this data with the data from conventional modeling (see Tab. 3 and Fig. 10), maximum time saving was 25 working hours or 78,13% (Variant 3, User 3) while minimum time saving was 11,00 working hours or 55,00% (Variant 1, User 5).

Biggest problem which occurred was with the drawings. Although, default section views, detail views, dimensions and scale are set, often operator had bigger problems to make functional drawings from them than from making new ones. Most common problem was the problem with predefined view positions. As this is set, as well as the default scale and, on the other hand, final model dimensions are differing, models were mostly outside the drawing format and dimensions were all around and messed up. This acquired a lot of work to do. Another point was the problem with some fonts. Changing from the default font type (ISOCPEUR) to any other font (Arial, Times New Roman) resulted in program crash. This is obviously some programming bug and it is a good example of earlier mentioned problems with custom-made software.

3. CONCLUSIONS

By building the variable model and GUI for the specific case, testing showed that the time needed for building the functional 3D model and 2D technical drawings for each part and sub-assembly decreased from 3,5 working days to roughly one work day. Time for adapting to the new sub-system and design rules weren't observed in this paper additionally but data collected shows that the complexity of the model has less influence on the time than number of use for each user. Obviously, there is the earlier mentioned issue about the first-time experiences.

Therefore it is expectable that during further exploitation the time will have decreasing tendency. Further on, it is not excluded that this kind of subsystem could be used by non-CAD specialists or engineers but by i.e. sales person. In this case, there is a possibility to use/adapt CAD variable models in other means than engineering. In the design process, it would definitely lead to reduction in design time. As this approach also minimizes the possibility to make an error for a designer and that it ensures company rules are fulfilled, if it would be uses as a framework for the company design team, it will also result in whole engineering process optimization.

Before implementation of this kind of a program, all the bugs must be fixed and company must ensure that the program is really working as it is supposed to.

Future research will include sub-program optimization, 2D technical documentation variability and automatic variable model integration to parent assemblies as well as further time decrease.

ACKNOWLEDGEMENTS

My gratitude goes to all my work colleagues for their help, as well as to my company for financial support. Also, I must acknowledge the work of Mr. Šagolj and his custom-made variable model built under my supervision and assistance. This work became an invaluable source of inspiration for me.
I would also like to extend my deepest gratitude to all my family, especially to my wife and daughter. Without their encouragement, all of my work wouldn't be possible and wouldn't have sense for me.
Finally, on a very personal note, I am grateful to my

mom and dad guiding me along the road of life, all the way.

ORCID

Goran Kukec http://orcid.org/0000-0001-6076-8394

REFERENCES

[1] Adeli, H.: Advances in Design Optimization, Chapman & Hall, 1994.

[2] Berg, W. P.: Pro/Engineer Tipps und Techniken, Carl Hanser Verlag GmbH & Co. KG, 2006. http://dx.doi.org/10.3139/9783446408050

[3] Bojčetić, N.; Štorga, M.; Marjanović, D.: Product data and knowledge integration.

[4] Cash, P. J.; Hicks, B. J.; Culley, S. J.: A comparison of the behavior of student engineers and professional engineers when designing, Design Conference 2012.

[5] Mahajan, V.; Muller, E.; Bass, F. M.: New product diffusion models in marketing: A review and directions for research, Journal of Marketing, 1, 1990, 1–26. http://dx.doi.org/10.2307/1252170

[6] Onwubolu, G.: Computer-Aided Engineering Design with Solidworks, Imperial College Press, 2013. http://dx.doi.org/10.1142/p761

[7] Šagolj, A.: Creation of the Alpas® frame of the fire fighting vehicle, University of Zagreb, Faculty of mechanical engineering and naval architecture, 2012.

[8] Štorga, M.; Pavlić, D.; Marjanović, D.: Reducing Design Development Cycle by Data Management Within the Design Office, 13th International Conference on Engineering Design - ICED 01, Glasgow, 2001.

[9] Walter, E.; Pronzato, L.: Identification of Parametric Models from Experimental Data, Springer, 1994.