**INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07**
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

# RULE-BASED MATERIAL TOPOLOGY MODELLING OF COMPOSITE STRUCTURES

**Thomas Kormeier[1] and Stephan Rudolph[2]**
[1, 2]Institute for Statics and Dynamics of Aerospace Structures,
University of Stuttgart, Pfaffenwaldring 27, 70569 Stuttgart, Germany
kormeier@isd.uni-stuttgart.de, rudolph@isd.uni-stuttgart.de

## ABSTRACT

This paper presents a methodology for the conceptual design synthesis and subsequent automated structural analysis of the material topology in composite structures. The long-term goal of this approach is to obtain a conceptual design language which is able to generate a multitude of composite structure topologies along with an automated analysis. The design language constitutes a graph-grammar based synthesis approach, which is well adapted to the topological layout problems. Ultimately, by integrating the design language and the analysis processes into one design loop, the presented approach shall lead to an optimization tool which yields topologically and functionally optimized composite structure designs.
First layouts and analysis' of composite laminate structures which were generated using this approach shall be presented in this work, along with a brief description of how the methodology is currently applied to model existing structures and to generate new design topologies.

*Keywords: Topology modification, Design Languages, Composite Structures*

## 1 INTRODUCTION

Recent developments in CAD and FEM tools such as CATIA [1] and MSC.Patran [2] have led to computer supported layout tools for laminated structures. The engineer is able to manually create computerized model descriptions of complex laminates starting with a CAD geometry followed by the manual definition of each ply and stacking sequence of the laminate.
To date, automated optimization of laminates is mostly performed through layer angle and stacking sequence optimization [3]. Several optimization techniques (i.e. genetic algorithms, gradient based methods etc.) have been applied to reach valuable results [4]. These methods do qualify well for parametric optimization, while research in the field of automated topologic layout is still in the early stages.
Introducing design languages in the material topology layout addresses the issue of an automated topological synthesis of the ply definition process and stacking sequence generation.

### 1.1 Design Languages

Design languages are means to describe the product synthesis in such a way that compilers (like the Design Compiler 43® used in this work) are able to execute pre-defined design rules automatically. Design languages use syntactical combinations of building blocks to form an abstract design representation in form of a metamodel. The metamodel acts as centralized model repository, which contains all relevant parametrical and topological design information. The analysis models of the respective design domains (CAD, FEM, CFD etc.) are generated automatically from this information during an interpretation phase.

Design languages constitute the working interface for the engineer to automate the design process. Know-how and best-practice experience are formally coded into reusable design patterns. The formalisms of design languages are discussed in depth in [5] and [6].

## 2 RELATED WORK

Conceptual synthesis is one of the most fundamental and difficult task in engineering since a multitude of different concepts are to be generated and evaluated. Antonsson and Cagan discussed a large variety of synthesis approaches in [7]. The approach of using grammars to formalize synthesis methods was firstly introduced by Chomsky in the technical context of production systems [8].
Grammar formulations are used as production systems to achieve the goal of providing a formal and thus automated method of design synthesis for a specific engineering domain. According to Schmidt et al. [9], applying grammar-based design option generation requires designing, validating and coding a set of grammar rules. Rule-based grammars offer the advantage of computerized execution, storage of know-how [10] and reusability, acceleration and formalization of the design process.
Lindenmayer and Prusinkiewicz [11] presented an example for the implications of changes made in a string-based production system describing the branching topology of plants. Shape grammars, as discussed by Gips and Stiny [12] go one step further and use geometrical shape descriptions to generate designs. Shea, Cagan and Fenves presented a robust structural synthesis method that uses shape grammars for the generation of truss structures [12]. Agarwal and Cagan proposed a coffeemakers shape grammar to replicate the geometry of many already existing designs and to propose new coffee maker forms [13]. Soman, Padhye and Campbell discussed a shape grammar approach that represents a variety of cutting and bending operations to construct feasible sheet metal shapes [14].
On a more abstract level, graph grammars use a graph as abstract design representation. In contrast to shape grammars, the rules do not operate geometrically, but combine basic building blocks (i.e. graph nodes) to form the design graph as metamodel description of the design. Schmidt et al. presented a general graph grammar methodology for the synthesis of kinematic mechanisms [9]. Campbell et al. applied grammars to define feasible function structures where the nodes represent functional entities and the rules combine them to form functional working principles [15]. Parallel grammars, based on a function-behaviour-Structure, are applied by Starling and Shea for the parametric synthesis of mechanical systems [16]. The Design Compiler® 43 [17] is being developed by the IILS mbH and the Institute of Statics and Dynamics at the University of Stuttgart. It provides the means to define engineering graph grammars that are referred to as design languages. As a novelty to previous attempts, the underlying design representation is domain independent. A methodology integrating both systematic design approaches and explorative bottom-up design procedures for the definition of design sequences (i.e. production systems) in design languages was presented in [6] regarding an aircraft design case study. Other applications of design languages describe the synthesis of satellites [18], space frame structures [19] and space stations [5] among others.

## 3 AUTOMATED DESIGN SYNTHESIS

In a design process it is common practice that the design is generated in accordance with a list of requirements which it must fulfil. If this process makes use of computer based modeling and analysis tools, an engineer may use the whole scope of functions offered by these programs. However, manually applied changes to the models may lead to very complex and time consuming operations. This problem is addressed by the introduction of an automated design synthesis process, which makes use of the scripting features of programs like CATIA and MSC.Patran.
The approach consists in identifying and capturing recurrent tasks and defining stereotypical design rules as prototypes for these tasks. By executing these design rules the design process is formalized and automated and leads to a metamodel design representation. Figure 1 shows the complete design loop in an automated design process using a design language as synthesis method.

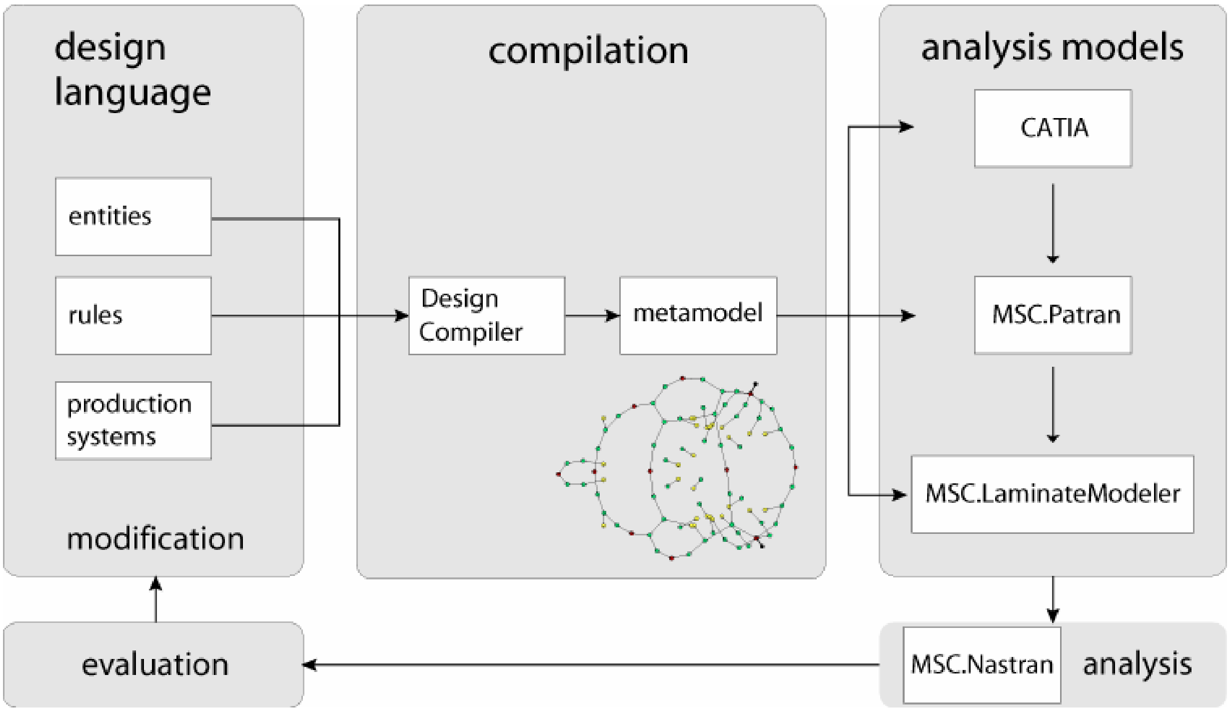


*Figure 1: The automated design process*

### 3.1 Decomposition

The metamodel shall be composed of basic entities which are to be combined to form the design representation. In order to reflect the properties of the design under consideration, these basic entities have to be identified and defined. Generally, it depends only on the degree of abstraction and the field of application how these entities are chosen. In a geometrically driven design, one would certainly consider basic geometric concepts like points, lines and surfaces to be stored in such building blocks, while a synthesis of laminate structures requires more abstract entities such as plies, materials and laminate properties. Even abstract concepts can be reflected by a building block in order to satisfy physical and functional requirements. The action of breaking a design up into its basic building blocks is called decomposition and is considered to be the first step into automation of a synthesis process.
Design languages as they are implemented in the Design Compiler 43® use building blocks as basic entities of the metamodel abstraction. These building blocks can include any of the following information concepts:

- Parameters and Variables
- Equations
- Associations to Files and CAD-Parts
- Interfaces to analysis programs in form of scripts

### 3.2 Design Rules

Based on the decomposition of the design, it is possible to define rules which capture tasks, actions and knowledge and thus allow a semantically and syntactically correct combination of the building blocks to form the metamodel representation. In other words, a sequence of design rules generates a topological description of the design by combining and relating the building blocks with each other.
A design rule can be seen as stereotypical pattern that identifies certain structures in a metamodel and subsequently modifies them by adding, deleting or changing entities in the model. These rules are defined graphically in the Design Compiler 43®.

A design rule searches for a structure (using a subgraph isomorphism search) in the existing metamodel and if predefined constraints are mated, the rule is applied. Elements can be deleted, added or topological relationships modified by a design rule

### 3.3 Interpretation of the metamodel

In an engineering context, the step following the synthesis of a design is analysis. The model has to be evaluated regarding the given requirements. The rule-based synthesis results in a metamodel (also referred to as design graph) that stores the parametrical and topological description of the design. In order to make this information available to analysis programs, it has to be translated into the respective analysis models. Each building block of the metamodel stores specific data needed to build the analysis model. This data is consists mainly in parts of control-scripts that are combined according to the topology of the metamodel. In this way, the analysis programs receive a script from the metamodel that automatically constructs the analysis model. The information processing chain used in this work consists of the Design Compiler, where the design language builds the metamodel. From this model, a CATIA model is generated that is transferred to MSC.Patran, where the FE-analysis model is built and evaluated. Figure 2 shows how data is transferred from the metamodel, the design graph, to CAD and FEM.

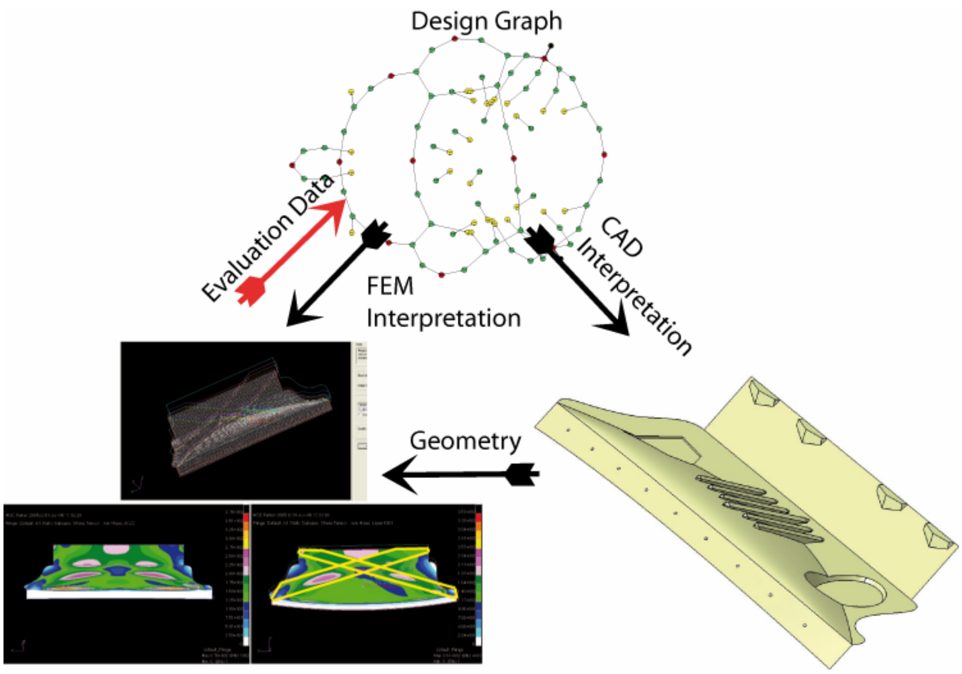


*Figure 2: Data flow within the processing chain*

After the analysis, it is possible to feed both knowledge based as well as classical optimization loops with the evaluation data. The knowledge based approach provides a set of design rules, that are supposed to react on specific patterns in the evaluation results, hence applying not only parametric changes but also topological modifications and engineering know-how available in the optimization loop.

## 4 DESIGN SYNTHESIS OF COMPOSITE STRUCTURES

The research focuses on the proof of applicability of a design language that integrates the geometrical definition of shell structures as well as the topological layout of a composite laminate based on the generated geometry. Existing topology optimization tools lean a workspace of finite elements, upon which boundary conditions are applied, until a residual structure is found that consists of elements that carry heavy loads. The results consist in a mass of finite elements that represent an optimized

structure. However, this structure still needs to be interpreted into a CAD-model. The approach followed in this paper aims at the generation of a component topology in form of a metamodel, which is optimized in its structural properties regarding manufacturing cost, material and feature layout and results in a valid CAD-model.
While a previous paper [21] discussed the topological layout of the geometry, the focus of the present work lies in the automatic generation of (topological) composite material properties based on these shell structures. The approach taken by the application of a design language enables us to quickly generate topological variants of composite components by applying some stereotypical design rules without the need to manually interact on CAD or FEM level. This procedure of planning and implementing a new design synthesis process shall be outlined in the following.

### 4.1 The concept

Since a material topology layout needs extensive information about the underlying geometry, the design language firstly generates this geometry, or rather its metamodel, which is translated into a CATIA-model. After extensive testing of various approaches to finite element modelling of laminates, we came to the conclusion that, since the analysis must be automated, the MSC.Laminate Modeler fits best into the planned processing chain. The Laminate Modeler receives the CATIA part as input and uses data directly from the metamodel to build the laminate plies. Boundary conditions are defined in the metamodel as well, where they are linked with their corresponding geometric entities, like bearings or force application points. Thus, by linking geometry, boundary conditions and material stacking properties directly in the metamodel, it is possible to automatically extract the data into a FEM model. It is however noteworthy that this whole process is strongly dependent upon the quality of the automatic meshing of the preprocessor, in this case MSC.Patran.
The design language is expected to build the topology of the laminate plies on top of the existing geometric metamodel. The rules are defined taking this into account and linking the plies with points and surfaces from the priorly defined geometry. Yet all this happens within the same design language, and hence not only material topology changes are possible in an optimization loop, but also modifications in the underlying geometry, should this be necessary. Since the geometry synthesis process was presented in a previous paper, it shall only be outlined in the following in order to give a better understanding of the general context of the work.

### 4.2 Decomposition of a composite structure

To generate the geometric description, the design language contains building blocks that represent geometric entities. These represent very basic concepts (points, lines, curves and surfaces) from which first a wireframe, then a basic surface model and at last the complete structure are generated. This structure is the middle-surface upon which the laminate is defined. In order to generate the laminate we need to introduce further entities, that allow us to control MSC.Patran and the Laminate Modeler, build a metamodel that fully describes the laminate and export the data to the preprocessor. Figure 3 how the laminate is defined and its connection to the geometric metamodel.

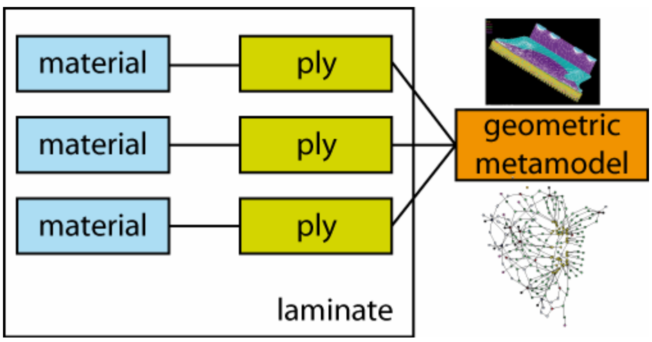


*Figure 3: Data structure linking the laminate layup with the geometry*

Each ply gets information about the underlying geometry and thus calculates its own geometrical properties. Furthermore, each ply has associated material data such as fibre directions, young's moduli and thicknesses to name a few. Since these plies are sequentially applied on the support surface, they are stored in a list structure and linked with the laminate.

### 4.3 Rule based laminate layout

The design rules are not discussed in depth for the sake of clarity, although the general principle should become clear. A production system that builds a laminate replicates the manual workflow in the preprocessor, while automating all tasks that do not require an "intelligent" input, as for example the information where to place a ply, how thick it should be and what orientation the fibres should have.

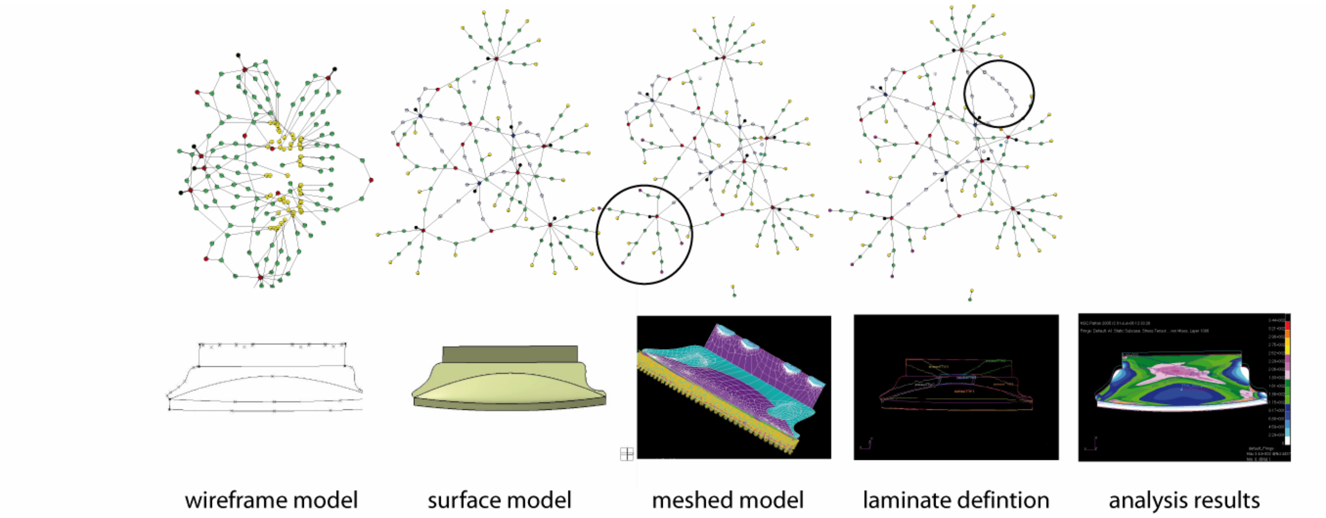


*Figure 4: Stepwise application of the production system*

Figure 4 constitutes an illustration of the stepwise application of the production system. To begin with, the geometric part of the production system builds a wireframe topology and from thereon a surface model, as described in [21]. This surface model is then translated into a CAD-model, which is imported in MSC.Patran where the meshing parameters from the metamodel and the boundary conditions are applied. In figure 4, the boundary condition nodes that were added are highlighted in the model above the meshed picture. The design rule associates the boundary condition with the geometry. Then the material properties are read from the metamodel and applied to the FEM-model. At last, the stacking of the plies is performed incrementally. A design rule positions the ply and gives it its properties. These properties can either be manually entered into the rule, or they are automatically derived from the associated geometry. The ply nodes can be discerned in the highlighted area above the laminate definition picture.

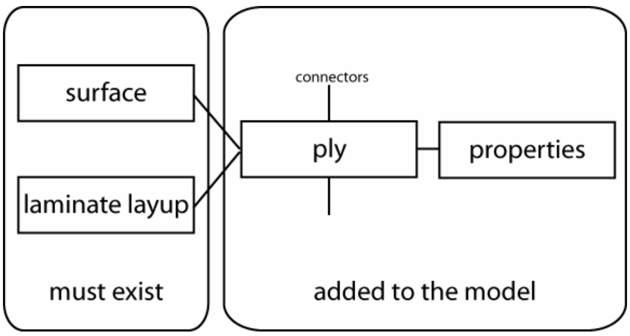


*Figure 5: The design rule to add a ply to the laminate*

Figure 5 shows the design rule, that adds a ply to the model. As constraints, both a surface to apply the ply to and a laminate must already exist in the metamodel. The ply and its properties are then added to the model. Other plies can be added via the connectors.

### 4.4 An example

The example in figure 6 shows a separator wall of a vehicle that is situated between passenger compartment and gas tank. Several unidirectional reinforcement plies have been placed upon the base geometry. These plies roughly follow a previously calculated load flow that results from a shear force.
Note that this model constitutes a test case and is not yet optimized in a real world context. The next step is refining the model by taking manufacturing and cost constraints into account in order to be able to generate a valuable conceptual design proposition.

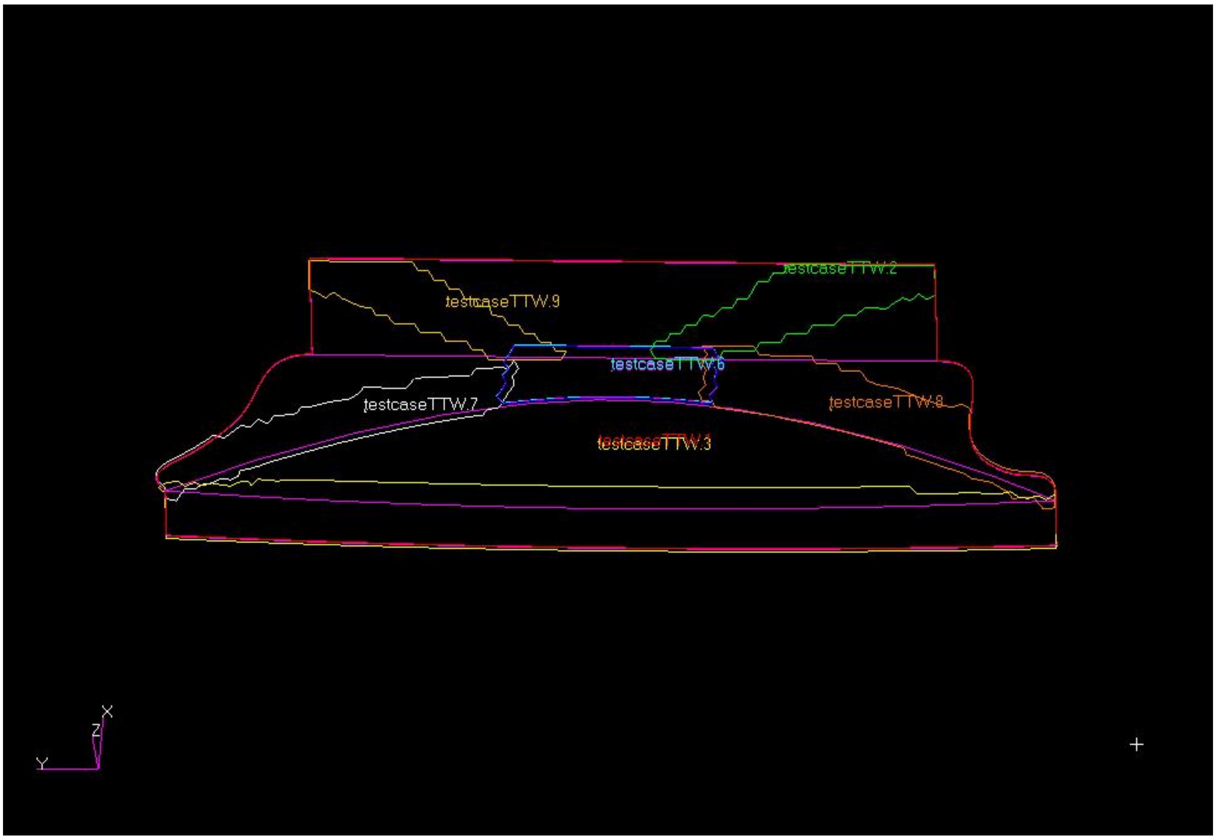


*Figure 6: Separator wall with reinforcement patches*

### 4.5 Advanced knowledge based design

Several strategies can be thought of in order to formulate design rules, that intelligently modify a given structure in order to adapt the laminate to a calculated load flow.
In analogy to the CAIO method [], it appears promising to calculate the load flow on the structure for an isotropic material. Based on this load flow, which is the result of the integration of the stress vectors, it is possible to apply for example unidirectional fibre reinforced patches that follow the load curves, as shown in the picture. A design rule takes the spline curve as input and calculates the start and end positions for each patch, as well as the dimensions for the patch and the fibre direction it must have in order to be aligned to the load flow. Figure 7 shows the spline and some reinforcement patches that are automatically placed by this rule

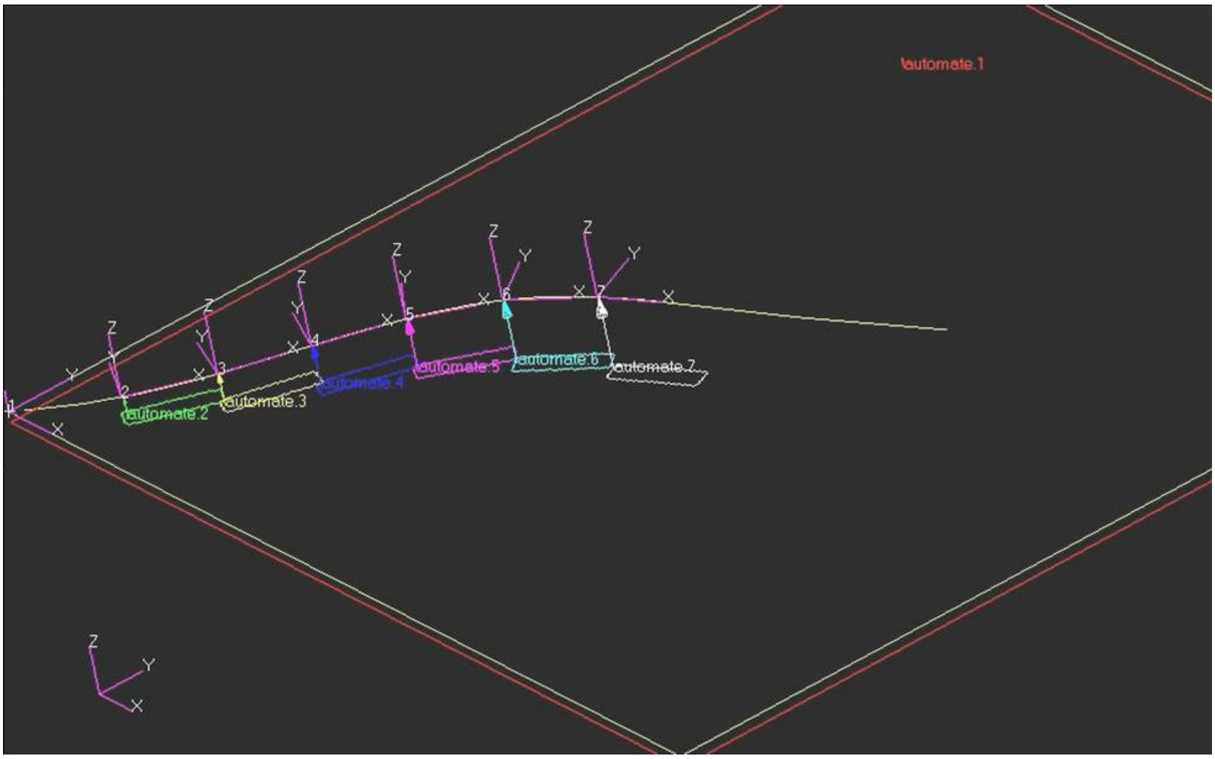


*Figure 7: Placement of reinforcement patches along a spline (load flow)*

This is a nice example for the association of FEM and geometry data. Moreover, it should be possible to extract the control data for the patch placement robot directly from the metamodel, and in so doing not only generating and optimizing the design, but also generating manufacturing information. Figure 8 shows how patches are placed along the load flow on a plate with hole.

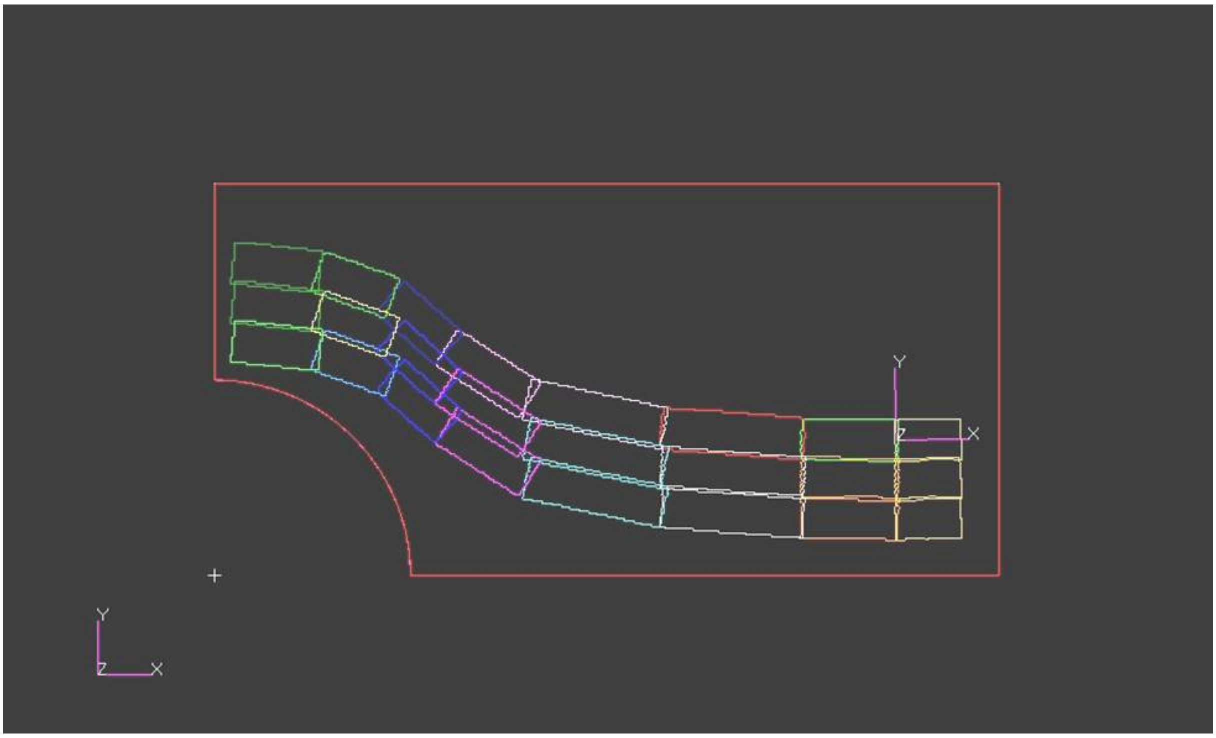

*Figure 8: Patches placed along the load flow on a plate with hole*

In a knowledge based optimization rule, it is expected that a rule can identify areas on the structure, where reinforcements are needed. The actual identification of these characteristic stress situations is the main focus of research for the future.

## 5 EVALUATION

The FE-analysis is realized with MSC.Nastran, using the input deck that was generated by the design language, MSC.Patran and the Laminate Modeler. To summarize, the CATIA model that is generated out of the metamodel is imported in MSC.Patran and the boundary conditions are applied. The metamodel stores all data that is necessary for the FE-analysis, including element type data, material models, meshing parameters, load data and evaluation data. This procedure does rely on the automatic meshing feature of MSC.Patran, which yields acceptable results as long as element sizes do not vary along large scales. Figure 9 shows an exemplified von Mises stress analysis of the gas tank separator wall without stiffening elements.

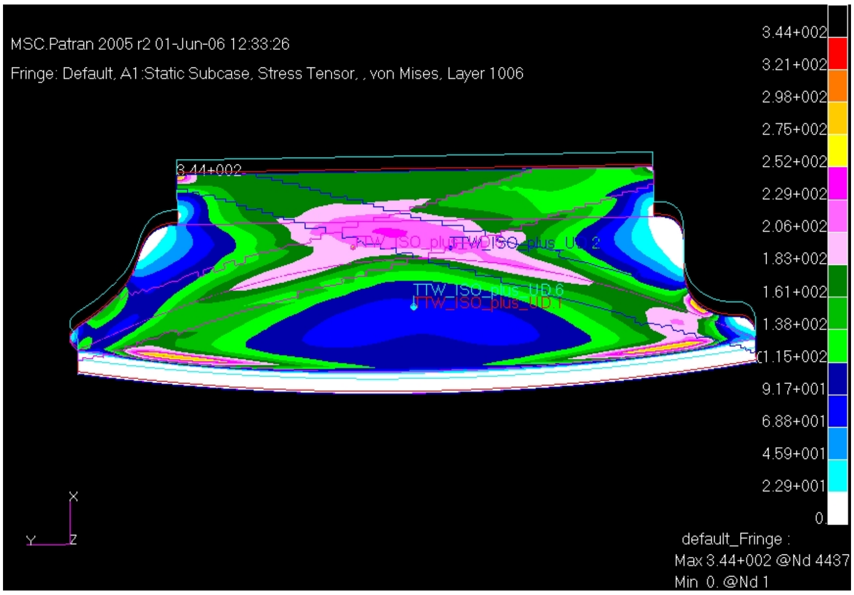


*Figure 9: Analysis of the separator wall*

The result file contains a field of stress data, which are associated with their corresponding finite elements. The metamodel however is made of discrete entities, which are not correlated with the finite elements. If data is to be transferred back from the result file into the metamodel, it is necessary to filter the stress field in order to write discrete values into the discrete entities of the metamodel. This shall be realized by a filtering layer, that shall also perform the task of identifying characteristic stress situations.

Regarding the formulation of knowledge based optimization rules, it is quite obvious that for a one-dimensional case this is not a difficult task. For the example of a simple bar, this could take the form of extracting a maximum strain and increasing the moment of inertia of the bars’ cross section. For a more-dimensional case it is not that obvious how to react on certain stress situations that establish themselves for a given boundary condition. A promising approach appears to be the identification of characteristic stress flows using principles from similarity mechanics as described in … If these characteristics are predefined, it should be possible to define a set of rules that modify the structure according to the stress field. This is however subject to further research. The goal is the computerized stress adaptation process that operates on the abstract geometric features in the design languages and thus results in a valid CAD-model.

## 6 CONCLUSION

A design language for the topological design of shell structures was enhanced with production systems for the topological description of composite laminates. The integration of the complete design process into an automated synthesis tool including geometry definition, material layout and subsequent analysis was achieved. The metamodel of a geometry is generated by rule-based synthesis of basic geometric entities and represents a surface-model of the design. This model is translated into a CATIA design which can subsequently be used in MSC.Patran and the laminate modeller to define the analysis model, including composite materials and reinforcements. Specific design rules process the geometric metamodel and sequentially add plies to the surface model, which ultimately leads to a definition of a laminate metamodel, which is directly translatable into the FEM preprocessor model. The use of a rule based approach offers the advantage of automated topological synthesis as well as modification of structure elements, which can be used for a knowledge based optimization.
Recapitulating, the parametric modifications of the composite structure design language include varying feature defining measure concepts, material properties and orientations. Furthermore, the following topological modifications are possible:

• Modification of the geometric topology
• Application of discrete reinforcement patches
• Definition and modification of the laminate layup
• Application of knowledge based variations

A case study for composite separator wall was generated with the design in order to prove the applicability of the approach. The benefit of the presented methodology lies in the reusability and automation of the design synthesis process, that will be incorporated into an automated topology optimization loop that yields a valid CAD-model as result. Furthermore, the definition of the laminate layup by design rules is a much more comfortable and intuitive way of building the analysis model.
The path towards a knowledge-based optimization rule system was discussed, including further work in the field of stress pattern analysis and rule based reaction schemes for certain stress situations on a component.
Although the presented design language exclusively generates surface structures, future efforts will be invested in the integration with other design languages to expand the scope of the approach in an engineering context.

## REFERENCES

[1] CATIA V5, Dassault Systemes, http://www.3ds.com/products-solutions/brands/CATIA
[2] PATRAN, MSC.Software, http://www.mscsoftware.com/
[3] Lin, C.-C., Lee, Y.-J., 2004, "Stacking sequence optimization of laminated composite structures using genetic algorithm with local improvement", Composite Structures, 63, pp. 339-345
[4] Gantovnik,V., Fadel,G.M., Gürdal,Z., 2006, „An Improved Genetic Algorithm For The Optimization Of Composite Structures", ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
[5] Rudolph, S., 2002, "Übertragung von Ähnlichkeitsbegriffen" (in German), Habilitation Thesis, Faculty of Aerospace Engineering and Geodesy, University of Stuttgart, Germany
[6] Kormeier, T., Rudolph, S., 2005, "On Self-Similarity As A Design Paradigm", Proceedings of Design Engineering Technical Conferences, ASME, Long Beach, California, USA
[7] Antonsson, E. K., Cagan, J., 2001, "Formal Engineering Design Synthesis", Cambridge University Press
[8] Chomsky, N., 1957, "Syntactic Structures", Mouton, The Hague, Netherlands
[9] Schmidt, L.-C., Shetty, H., Chase, S., 2000, "A Graph Grammar Approach for Structure Synthesis of Mechanisms", Journal of Mechanical Design, **122**, pp. 371-376
[10] Shah, J. J., Mäntylä, M., 1995, "Parametric and Feature-Based CAD/CAM", John Wiley &

Sons Inc., New York, NY, USA
[11] Lindenmayer, A., Prusinkiewicz, P., 1996, "The Algorithmic Beauty of Plants", Springer Verlag, Berlin, Germany
[12] Shea, K., Cagan, J., Fenves, S. J., 1997, "A shape annealing approach to optimal truss design with dynamic grouping of members", Journal of Mechanical Design, **119**, pp. 388-294
[13] Agarwal, M., Cagan J., 1998, "A Blend of Different Tastes: The Language of Coffeemakers", Environment and Planning B: Planning and Design, **25**, pp. 205-226
[14] Soman, A., Padhye, S., Campbell, M., 2003, "Toward an automated approach to the design of sheet metal components", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, **17**, pp. 187-204
[15] Sridharan, P., Campbell, M., 2004, "A Grammar for function structures", Proceedings of Design Engineering Technical Conferences, ASME, Salt Lake City, Utah, USA
[16] Starling, A. C., Shea, K., 2005, "A Parallel Grammar for Simulation-Driven Mechanical Design Synthesis", Proceedings of Design Engineering Technical Conferences, ASME, Long Beach, California, USA
[17] Ingenieurgesellschaft für Intelligente Lösungen und Systeme, IILS mbH, http://www.iils.de
[18] Schaefer, J., Rudolph, S., 2004, "Satellite design by design grammars", Aerospace Science and Technology, **9**, pp. 81-91
[19] Haq, M., Rudolph, S., 2004, "EWS-Car: A Design Language for Conceptual Car Design" (in German), Proceedings of Numerical Analysis and Simulation in Vehicle Engineering, VDI-Reports 1846, pp. 213-237
[20] Irani, M. R., Rudolph, S., 2003, "Design Grammars for Conceptual Designs of Space Stations", Proceedings of International Astronautical Congress, Bremen, Germany
[21] Kormeier, T., Rudolph, S., 2006, "Topological Design Of Shell Structures By Design Languages", Proceedings of Design Engineering Technical Conferences, ASME, Philadelphia, PA, USA