A Knowledge Based Engineering Approach to Support Automatic Design of Wind Turbine Blades

T.G. Chiciudean[1] Gianfranco La Rocca[1] and Michel J.L. van Tooren[1]
[1]Delft University of Technology, Delft, the Netherlands www.dar.lr.tudelft.nl

Abstract

Multidisciplinary Design and Optimisation is the future design methodology with the largest potential for helping the engineers to push further the limits of design. Nowadays the design process is distributed between different teams of specialists on different geographical locations, facing difficulties associated with managing and integrating the design. To be able to face these challenges a new design environment addressed as Design and Engineering Engine, is introduced here. In order to automate the design process without constraining creativity and innovation, a parametric system called the Multi Model Generator is presented in this paper and exemplified with a case study on the UPWIND 5 MW reference wind turbine blade.

Keywords:

Multidisciplinary, Design, Integration, Knowledge Based Engineering.

1 INTRODUCTION

The wind turbine design of the next decade will integrate the newest discoveries of our days pushing the wind energy technology at limits in order to fulfil the market demand (e.g., high power output 10-20 MW with low cost, deep water installation up to 200 m, extreme wind conditions at low temperature etc.). Upscaling the power output by simply upscaling the dimensions of the wind turbine, without radical changes in the design, is beginning to reach its limits. The designer is forced to explore and integrate new technologies and design features like high power direct drive generators, floating support structure for deep water, blade flaps for loads reduction and stability control etc.

To be able to support the design and evaluation of these new wind turbines, a new design environment is in development. This system aims to cover the conceptual and preliminary design phases of a wind turbine. A schematic overview of these two phases is given in Figure 1.

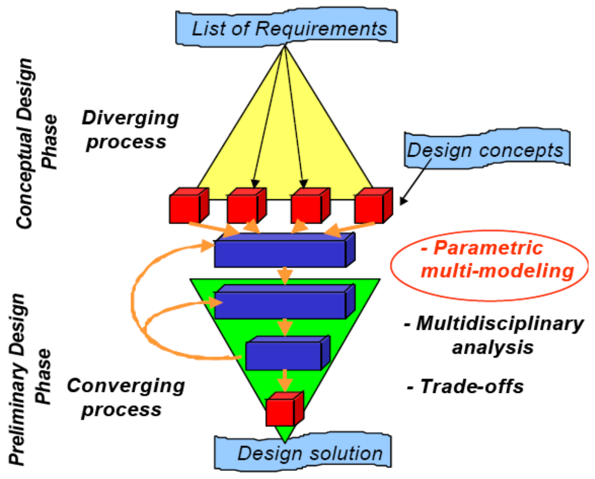


Figure 1: Diverging/converging design process

In the conceptual design phase, one or more concepts (a confirmed combination of working principles) are generated, concepts that could fulfil the customer requirements. In the preliminary design phase a transition is made from abstract conceptual ideas to the physical shape of parts and components of a product in order to be analysed and evaluated.

The design process is a highly interactive and time consuming process in which many repetitive tasks are performed [1]. Considering the cost and time pressure in the current market, a partial flexible automation of the process is highly recommended. An automation of the conceptual design phase is difficult to be made because here the creativity and the experience of the designer play an important role. But the preliminary design phase is usually more suitable for automation due to the repetitive natures of the analysis performed here.

In order to automate as much as possible both design phases and to capture the designer intent, a Knowledge Based Engineering (KBE) approach is appointed to develop the new design environment, addressed as Design and Engineering Engine (DEE). In the next section the philosophy behind the DEE concept and its structure and functionality are presented in more details.

2 THE DESIGN AND ENGINEERING ENGINE

A DEE is defined as an advanced design environment, where the design process of complex products can be supported and accelerated through the automation of non-creative and repetitive design activities. A DEE is typically multi-site (the components of the DEE are distributed over different physical locations) and multi-disciplinary [2].

In practice, a DEE consists of a collection of commercial of-the-shelf (COTS) components connected by a framework. This concept is illustrated in Figure 2.

The main components of the DEE are:

- **Initiator.** The initiator is responsible for providing feasible starting values for the instantiation of the (parametric) product model and, in some cases, can be an optimizer by itself.

- **Multi-Model Generator (MMG).** The MMG is responsible for instantiation of the product model and extracting different views on the model in the form of report files to facilitate the analysis tools.
- **Analysis Tools.** The analysis tools are responsible for evaluating one or several aspects of the design (e.g. structural response, aerodynamic performance, aero-elastic stability or manufacturability).
- **Converger and Evaluator.** The converger and evaluator are responsible for checking convergence of the design solution and compliance of the product's properties with the design requirements.
- **The framework.** A set of communication routines responsible for the connection between the various tools of the DEE.

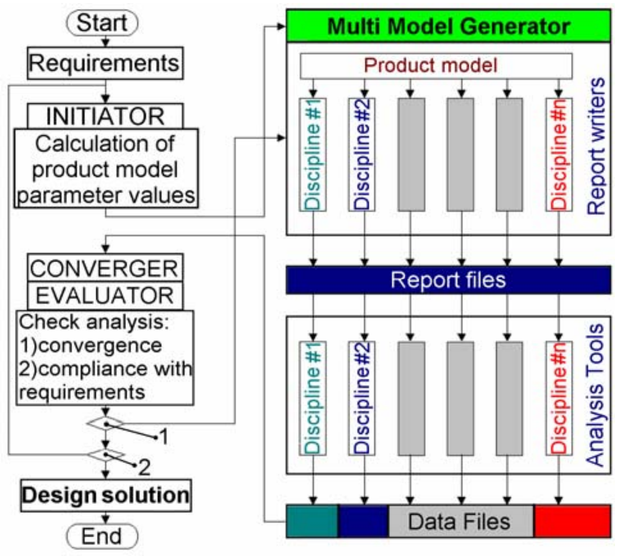


Figure 2: The Design and Engineering Engine concept

The definition of the product, the problem to be solved by the DEE, is based on High Level Primitives (HLPs). These are functional building blocks, which allow the user of the DEE to define a product in a certain product family. These functional blocks are basically sets of rules that use parameters to initiate objects that represent (part of) the product under consideration or to apply an engineering process to the initiated object [2]. So far three HLPs have been developed: the blade-trunk, nacelle-trunk and connection element, which allow the generation of different wind turbine configuration (see Figure 3).

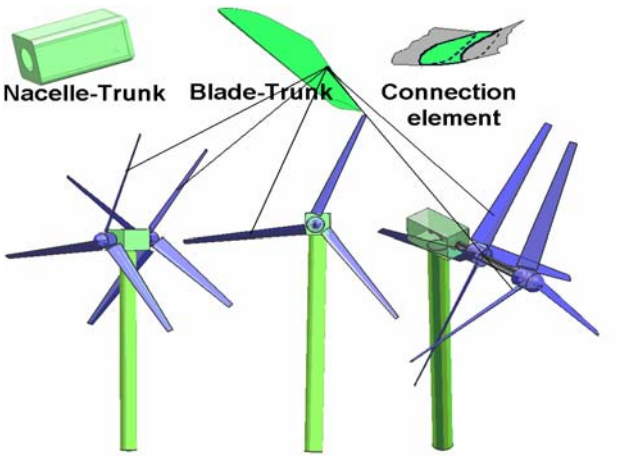


Figure 3: Basic HLPs defining different wind turbine configurations

The parameters and variables inside the HLPs are addressable by basically any program and therefore these functional blocks can be used in analysis and optimization loops. The concept of the HLPs is implemented using a KBE approach instead of traditional CAD. KBE allows the definition of product models based on product structure and rule bases for the selection and instantiation of components within that structure. When properly used, the resulting KBE-model allows extraction of different geometrical and non-geometrical views on the product after instantiation of the model with a specific set of parameter and variable values. The objects that are responsible for this are called capability modules and are mixed-in in the HLPs.

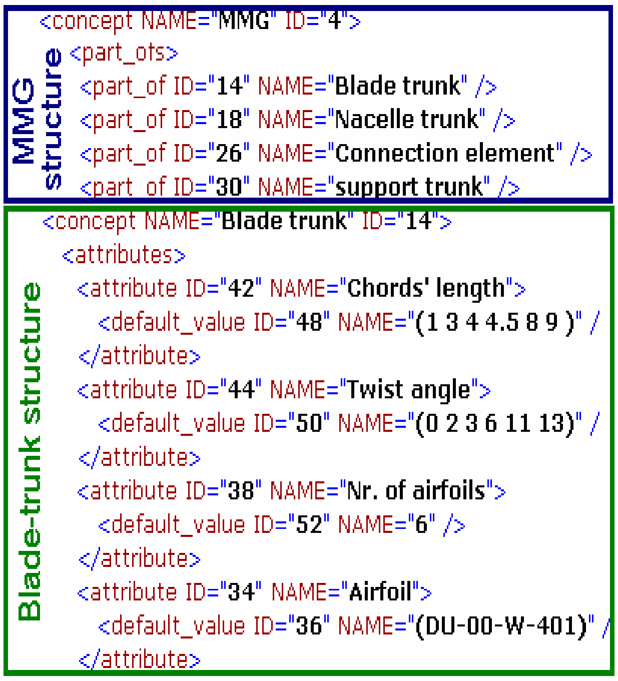
For a better understanding of the HLPs concept, the implementation of a blade trunk HLP inside the MMG by means of a KBE system is presented in the next section.

3 DEVELOPING A MULTI-MODEL GENERATOR FOR UPWIND PROJECT.

UpWind is a European project funded under the EU's Sixth Framework Programme (FP6). The project looks towards the wind power of tomorrow, more precisely towards the design of very large wind turbines (8-10MW), both onshore and offshore [3].

UpWind focuses on design tools for the complete range of turbine components. It addresses the aerodynamic, aero-elastic, structural and material design of rotors [3].

The DEE under development for UpWind project consists, in this case, of an aggregation of design, modelling and analysis software tools, (both commercial off-the-shelf and in-house developed by the various UpWind partners) properly interconnected by means of a software communication framework [4]. To control the data information flow inside the framework a Knowledge Management (KM) environment using PCPACK has been employed to support the design process. PCPACK is a set of tools that helps the designer to identify, capture and store the knowledge in a XML file or graphic views. Inside the XML file each concept, relation or attribute is univocally defined. An example of a XML file created using PCPACK for the MMG and blade trunk structure is presented in Figure 4.



```
<concept NAME="MMG" ID="4">
  <part_ofs>
   <part_of ID="14" NAME="Blade trunk" />
   <part_of ID="18" NAME="Nacelle trunk" />
   <part_of ID="26" NAME="Connection element" />
   <part_of ID="30" NAME="support trunk" />
<concept NAME="Blade trunk" ID="14">
  <attributes>
   <attribute ID="42" NAME="Chords' length">
     <default_value ID="48" NAME="(1 3 4 4.5 8 9 )" /
   </attribute>
   <attribute ID="44" NAME="Twist angle">
     <default_value ID="50" NAME="(0 2 3 6 11 13)" /
   </attribute>
   <attribute ID="38" NAME="Nr. of airfoils">
     <default_value ID="52" NAME="6" />
   </attribute>
   <attribute ID="34" NAME="Airfoil">
     <default_value ID="36" NAME="(DU-00-W-401)" /
   </attribute>
```

Figure 4: XML view of the MMG and blade trunk structure

In order to comply with the functionalities required by the DEE the blade trunk HLP has to fulfil the next requirements:

- Definition of a full parametric description of the blade.
- No limits and constraints to the creativity of the design.
- Transparency of the blade model generation process.
- Software structure modularity.

The GDL environment has been selected for development of the MMG to serve the DEE. The MMG basically consists of a set of routines programmed in GDL, which is a super set of LISP, routines that are the physical representation of the HLPs concept. GDL is an advance tool to support KBE and is Object Oriented software that contains both the typical features of expert system languages and the geometrical handling possibilities of advance CAD programs.

The model created using GDL is not a simple CAD model; it is a generative model capable to represent the engineering intend behind the geometric design. It captures the **How** and **Why** in addition to the **What,** of the design. It captures the design strategy required to produce a particular product from a specification. It is the set of engineering rules (not only rules involving geometry) used to design the product [5].

The definition of a blade trunk requires a set of parameters that univocally defines the internal and external structure of the blade. In this particular case the starting values of the parameters are collected from the UpWind 5MW reference wind turbine for the first instantiation [3]. A list of parameters used to define the blade trunk is listed below:

- Type of airfoils (selected from a predefined-but-updateable library)
- Amount of airfoils (minimum 2 airfoils, the value of this parameter is computed automatically by the MMG it represents the length of the list for inputting the other parameters)
- Pitch axis (represents also the reference axis to apply twist)
- Twist angle
- Rotor radius (used for positioning the airfoils along the pitch axis)
- Chords' length of airfoils
- Thickness of airfoils

For better understanding of the structure of the blade trunk and how these parameters are defining the geometry a graphical representation of a blade section is presented in Figure 5.

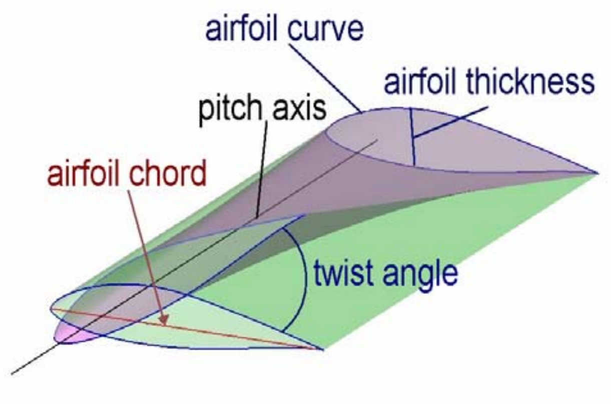
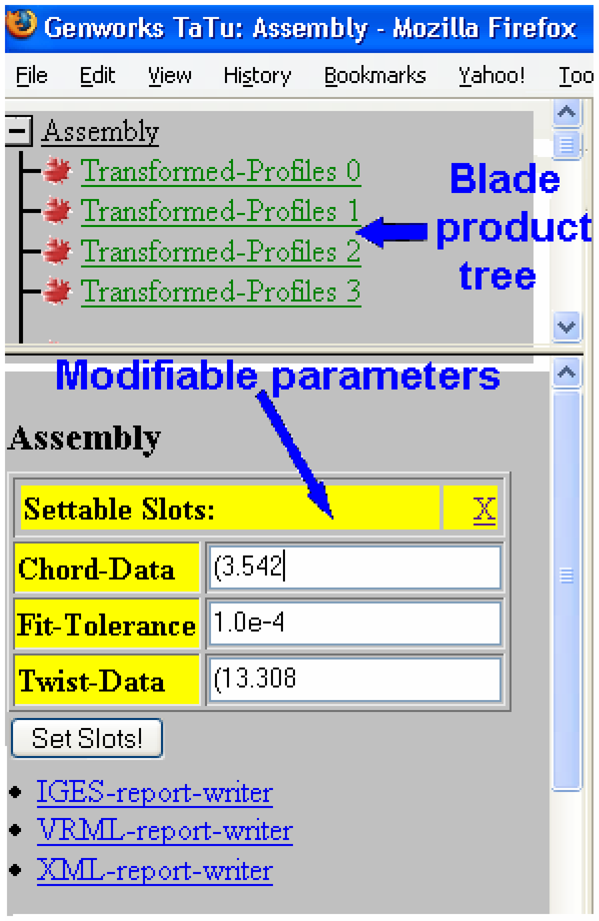


Figure 5: Blade trunk definition

For inputting the parameters inside the MMG the user has in hand three options. Depending at what stage it is in the design process the user can choose between direct input file and KBE dynamic input. For direct input file the parameter can be edited in a text file or structured in PCPACK in a XML file (see Figure 4), these two options are available in all the design stages. The KBE dynamic input option is available using the settable slots field via a web graphical interface developed in GDL after the first instantiation of the model, instantiation that requires one of the first input options. Using the web graphical interface the designer can inspect the geometry and verify the design boundaries in terms of dimensions conflicts and components positions without modifying the model structure. This feature allows the user to avoid bottlenecks on hard coding where is more difficult to identify a dimension conflict or components position errors. All modifications made using this interface are just locally in terms of view updates. A graphical web interface developed using GDL showing the blade structure tree and settable slots section for modifying certain parameters in an instance is presented in Figure 6.



Chord-Data	(3.542
Fit-Tolerance	1.0e-4
Twist-Data	(13.308

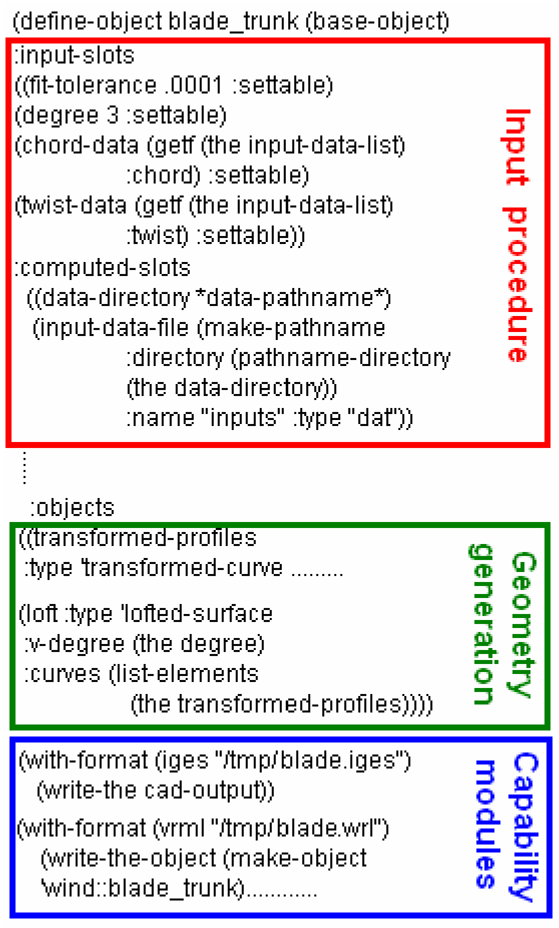
Figure 6: KBE graphical development interface

In order to perform an optimization loop inside the DEE the input procedure coded in the MMG is based on functional programming. Functional programming is a programming paradigm that treats computation as the evaluation of mathematical function and avoids state and mutable data [6]. Whenever a modification appears in the optimization the blade trunk is automatically recomputed without any user interference.

The blade trunk geometry is generated by interpolating a surface through a set of defined planer curves (the airfoil curves). Before the interpolation all the curves are scaled, positioned and rotated according to the provided input values.

After the instantiation of the model the user can extract different views in terms of geometry output on different formats IGES or ASCII, CFD points, mass distribution, and mass momentum of inertial distribution. For outputting these views the user can use the web graphical interface or directly recompiling the capability modules from the MMG. In an optimization loop this views are automatically regenerated and outputted whenever the DEE is requesting.

A code exemplification showing the input procedure, geometry, generation procedure and the capability modules is illustrated in Figure 7.



```
(define-object blade_trunk (base-object)
:input-slots
((fit-tolerance .0001 :settable)
(degree 3 :settable)
(chord-data (getf (the input-data-list)
            :chord) :settable)
(twist-data (getf (the input-data-list)
            :twist) :settable))
:computed-slots
 ((data-directory *data-pathname*)
  (input-data-file (make-pathname
            :directory (pathname-directory
            (the data-directory))
            :name "inputs" :type "dat"))
  ⋮
 :objects
((transformed-profiles
 :type 'transformed-curve .........
(loft :type 'lofted-surface
 :v-degree (the degree)
 :curves (list-elements
           (the transformed-profiles))))
(with-format (iges "/tmp/blade.iges")
  (write-the cad-output))
(with-format (vrml "/tmp/blade.wrl")
  (write-the-object (make-object
  'wind::blade_trunk)...........
```

Figure 7: Blade trunk code exemplification.

The blade trunk model is capable to generate the blade geometry using up to one thousand profiles. For an accurate and smooth geometry the user can specify how many blade trunks to be used. The optimal number of blade trunks is defined in equation (1).

$$n_b = \frac{n_a}{2} \qquad \text{if } n_a \text{ is even number}$$

(1)

$$n_b = \frac{n_a - 1}{2} \qquad \text{if } n_a \neq 1 \text{ and odd number}$$

Where n_a is the number of airfoils type and n_b the number of blade trunks.

When more than one blade trunk is used to generate the blade geometry a set of connection elements is necessary to be used for handling the transition. A connection element has the same structure like the blade trunk; the difference is a special KBE routine that evaluates the smoothness of the transition surface. An illustration of a blade geometry generated using a multi trunks approach is presented in Figure 8.

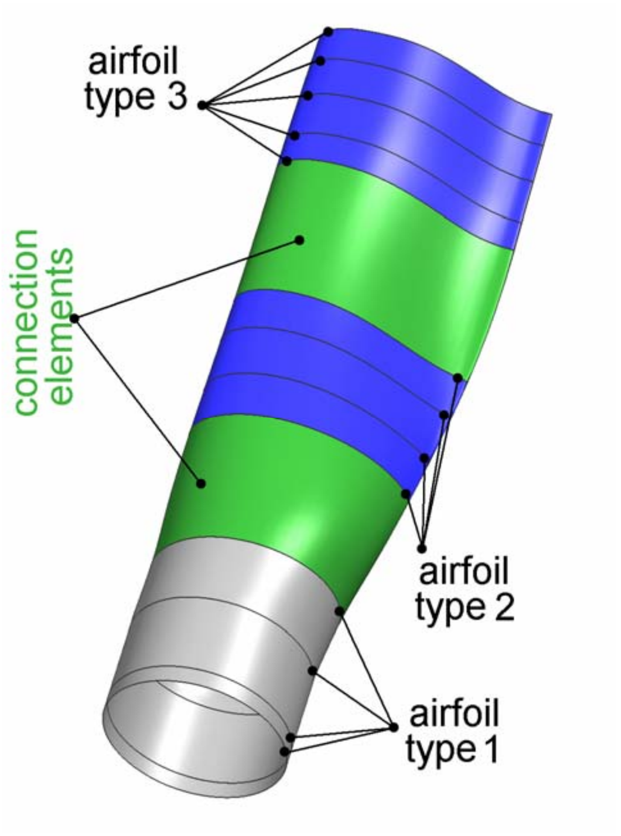


Figure 8: Multi trunks wind turbine blade section

The multi trunks approach for the geometry generation it's allowing the analysis tools to change the geometry of each section independently. This approach is the most suitable approach when the MMG is connected with a multi body dynamics code in an optimisation loop and this is one of the future targets of the MMG.

4 CONCLUSIONS

The implementation of a blade trunk HLPs inside the MMG has been presented. The future development of the DEE will include a set of new analysis tools and new capability modules have to be developed to comply with the functionalities required by the DEE.

The MMG will be aimed also to support studies of innovative rotor configurations for load alleviations. These will include the implementation of moveable elements for the leading and trailing edges (similar to aircraft flaps and slats), controlled by mechanical and/or piezoelectric actuators. The challenge of exploring in a relatively short time a large amount of so many different configurations will make of the generative capabilities of the KBE modelling system an indispensable ingredient to the project success.

For the generation of the entire wind turbine the blade-trunk, nacelle-trunk, structure-trunk and connection element are still in development and will include the new set of features mentioned in this paper.

5 ACKNOWLEDGMENTS

The study reported in this paper is carried out in the UpWind project funded under the EU's Sixth Framework Programme (FP6). The authors wish to thanks the EU for funding the project.

Thanks to Dave Cooper from Genworks International for the support given in the implementation of the MMG.

6 REFERENCES

[1] Cerulli, C, J.P.T.J. Berends, M.J.L. van Tooren and J.W. Hofstee, 2005, Parametric Modeling for Structural Dynamics Investigation, International Forum on Aeroelasticity and Structural Dynamics, CEAS/AIAA/DGLR.

[2] Tooren M.J.L. van, M. Nawijn, J.P.T.J. Berends and E.J. Schut, 2005, Aircraft Design Support Using Knowledge Engineering and Optimisation Techniques, *Structural Dynamics and Materials Conference*, *AIAA/ASME/ASCE/AHS/ASC Structures.*

[3] UpWind web page www.UpWind.eu

[2] Dave Cooper, La Rocca G., 2007,Knowledge-based Techniques for Developing Engineering Application in the 21st Century, Integration and Operations Conference, 7th AIAA Aviation.

[5] La Rocca G., L. Krakers and M.J.L. van Tooren, 2002, Development of an ICAD Generative Model for Blended Wing Body Aircraft Design, Symposium on MDO, Proceedings of the 9th AIAA/ISSMO.

[6] Graham, P., 1996, ANSI Common Lisp, Prentice Hall.