A KNOWLEDGE-BASED MASTER MODELING APPROACH TO SYSTEM ANALYSIS AND DESIGN

Marcus Sandberg[1], Ilya Tyapin[1], Michael Kokkolaras[1], Ola Isaksson[2]
(1) Luleå University of Technology, Sweden, (2) Volvo Aero Corporation, Sweden

ABSTRACT
The jet engine industry relies on product models for early design predictions of attributes such as structural behavior, mass and cost. When the required analysis models are not linked to the governing product model, effective coordination of design changes is a challenge, making design space exploration time-consuming. Master modeling (MM) approaches can help alleviate such analysis overhead; the MM concept has its origins in the computer-aided design (CAD) community, and mandates that manual changes in one model automatically propagate to assembly, computer-aided manufacturing (CAM) and computer-aided engineering (CAE) models within the CAD platform. Knowledge-based master models can also be used to communicate changes in the product definition to models that are external to the CAD platform. This paper presents details of the knowledge-based master modeling approach as applied to mechanical jet engine analysis and design, where different fidelity models and analysis tools are supported in the early design stages.

Keywords: Knowledge-based engineering, master model, multidisciplinary analysis and design optimization, jet engine

1 INTRODUCTION
The decisions made in the early phases of product development (PD) have a decisive impact on the product and its use throughout the life cycle. As a consequence, effective design requires modeling approaches that can predict the properties and behavior of forthcoming products in early PD stages. The Systems Engineering discipline provides overarching methods for how to approach such design problems systematically [1], but the quality of data available for evaluation is a bottleneck. A better and more precise description of the product and its environment is thus required. Using virtual product modeling techniques, such details can be generated by creating conceptual design solutions with a greater resolution than ever before.
Virtual product models are commonly used to predict life-cycle effects of alternative designs, with the aircraft industry being a driving force of development of computer aided design (CAD) and computer aided engineering (CAE) techniques. CAD and CAE technologies have enabled the creation of product models for digital mock ups, weight estimation, and rotordynamics, stiffness, fluid dynamics and performance analysis. The need of a number of analysis models for each discipline creates a coordination challenge for product changes since there is seldom a single product definition during early design stages. The synthesis of analysis results often leads to re-modeling and design iterations necessary to re-assess the behavior of products after the separate disciplinary design and simulation activities have been conducted. Many analysis models and product representations have to be created several times, and merely the co-ordination of these modeling activities tends to be a costly and time-consuming exercise (see Figure 1).
A master model (MM) approach means having one managing model to control other models. Once the master model is changed, then the associated models are updated accordingly. One of the first master model approaches was reported by Newell and Evans [2]. Within the CAD field MM technology is more or less taken for granted but within the CAE field MM technology is less established. There exists a lot of work on multidisciplinary analysis, but not much of it focuses on how to manage product definition changes that occur when domain-specific models are used concurrently for different analyses. This makes it challenging to conduct design optimization. Commercial software environments such as iSIGHT, Optimus, ModeFRONTIER or ModelCenter provide techniques to link one product definition to different models. To provide more flexibility in geometry change and analysis model linking compared to traditional parametric CAD, knowledge-based engineering can be

employed to control the MM. Previous work has demonstrated that it is possible to use KBE to create MMs [3, 4]. However, there is a need to further detail the actual use of the KBE technology to create the MM. This paper provides details about this process by means of a whole jet engine model.
The paper is structured as followed: In section 2 a research background is given to present research within the fields of knowledge-based engineering, CAD-CAE integration and knowledge-based master-models. Section 3 explains the knowledge-based MM approach and section 4 presents an application of this approach. Section 5 wraps up the paper with concluding remarks.

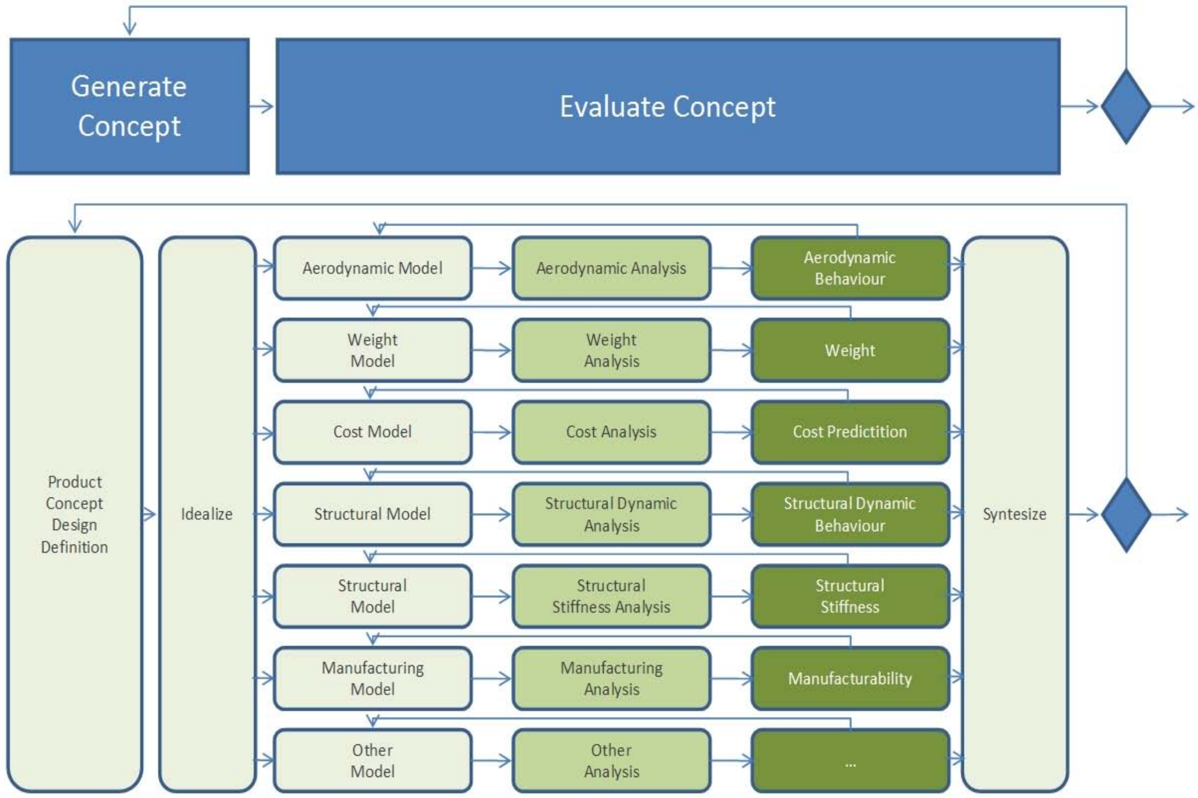

Figure 1. Conceptual design and analysis using unlinked product models

2 BACKGROUND

Automating chains of engineering tasks has been the approach of experienced engineers since the dawn of the computer. Knowledge-based engineering stems from knowledge-based systems, [5], and is claimed to have been coined at the release of the CAD software iCAD [6]. Stokes defines KBE as *"the use of advanced software techniques to capture and re-use product and process knowledge in an integrated way"* [7]. The core is about creating a generative model that can generate product development items such as geometry, reports, BOMs, or finite element models [8]. By using rules, geometry objects can be modeled in a way beyond traditional parametric models. Radical topological changes, e.g. changing a cylinder into a rectangular prism, are possible. For routine engineering tasks KBE applications were found useful [9, 10]. During the last decade, the major CAD/PLM vendors have adopted KBE modeling capabilities in, for example, Siemens NX and Dassault Systemes CATIA.
There exist numerous approaches where the challenge of integrating CAD models with CAE models is targeted. Lee presented a CAD-CAE (computer-aided engineering) integration strategy for feature-based design [11]. The strategy is based on a MM that creates the required CAD and CAE models. CAD model creation is done interactively with the user. The abstraction and dimensional function is semi-automatic. Since the Lee framework is not fully automatic, further work is needed to use it in an optimization loop. Hong-Seok and Phuong integrated CAD and CAE using scripts, programming languages, application programming interfaces and meta-modeling to perform structural optimization [12]. Their approach is limited to traditional parametric capabilities; more radical geometry changes, permitted by KBE, are absent.

In the field of master modeling techniques Hoffman and Joan-Arinyo suggested a master model architecture centered around a server and a repository to which different clients can connect to [13]. These clients can be CAD systems, geometrical dimensioning and tolerancing agents, manufacturing process planners or other downstream clients. Each client receives their view of the design. Each design change made by one of the clients causes changes to other clients' views according to a change protocol and permissions. The architecture is semi-automated and user interaction is needed. La Rocca and van Tooren presented a framework to enable MDO supported by KBE [3]. The core unit of the system consists of a multi-model generator (MMG) that can generate numerous aircraft component (exemplified with an aircraft wing) configurations based on a high-level primitives concept. The MMG can extract data and information from the product definition to specific analyses. Design (product definition) changes are propagated in an automated fashion to all analysis models. A toolbox checks the analysis convergence and compares results with the design specification. If failing to satisfy the specification, the toolbox can trigger new design iterations.

Despite the relative success of KBE approaches proven, the design methodology of using a governing master model is not well established. Best practice that maximize the use of software functionality offered by vendors implies a risk over time where methodology and rule dependencies may become obsolete as new software tools and versions are launched. A system-independent, yet system-implementable, design logic is needed. Detailing the actual constituents of the knowledge-based master model (KBMM) is of primary interest. It is also of interest to develop KBMMs for jet engine structures, as exemplified in this paper, in order to complement the application presented in [3]. The investigation of implementation issues in KBE software environments other than ICAD, which was used in [3], is also of interest.

3 THE KNOWLEDGE-BASED MASTER MODEL APPROACH

This section explains the knowledge-based master model (KBMM) approach in terms of the general idea and its constituents. The general idea of the KBMM approach is to use a CAD system and its KBE software to link all analysis models to one centralized product definition so that early product development can be made more effective. When new ideas need to be tested the product definition can be changed and these changes are automatically propagated to the linked models. An additional goal is to completely automate the design and analysis activities, so that the managing unit can be used to handle the optimization process. By using the capabilities of KBE software within a CAD system it is possible to further enhance the master-model ideas since KBE can enable more flexible geometry configuration compared to traditional parametric CAD. The rules within the KBE classes are also suitable to be used to link the analysis models to the governing product definition.

The KBMM contains of a managing unit, KBE classes and API (Application Programming Interface) calls and Macros as shown in Figure 2.

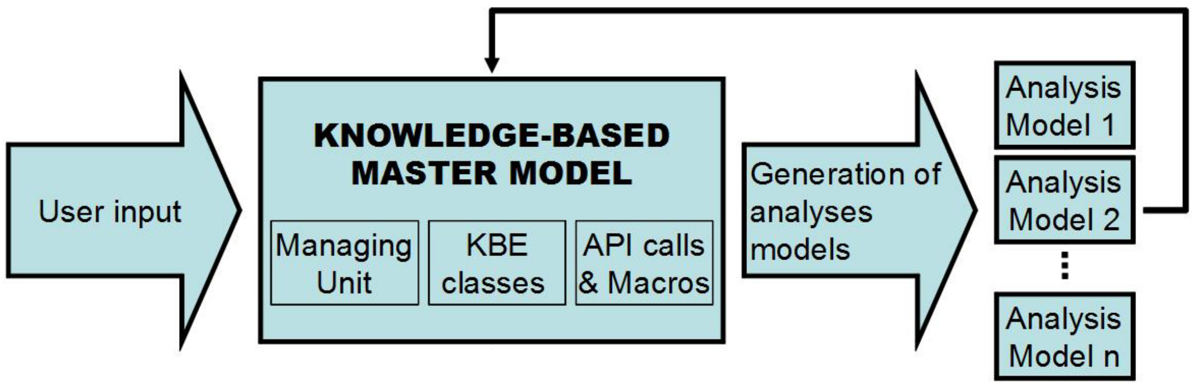

Figure 2. Overview of the knowledge-based master model approach

3.1 The managing unit

The managing unit coordinates the design and analysis loop by taking the user input and initiating required models and analysis activities. Since KBE software is often coupled to the CAD software the managing unit is needed to either operate through the API or the CAD software or in another way to be able to fire the instantiation of KBE classes, API calls and Macros. Optimization functions are included into the managing unit.

3.2 The KBE classes

The KBE classes can generate geometry, finite element objects (e.g. mesh, boundary conditions) and geometry analyses (e.g. weight calculation). Object-oriented KBE software (such as Knowledge Fusion in Siemens NX) usually has predefined classes for fundamental geometry objects (e.g. block, cylinder, ellipse, datum-plane) and predefined methods or functions for parameter handling (e.g. max, min, floor, sin) and CAE operations (e.g. meshing, boundary conditions, loads). The rules that govern the analysis model generation reside within the user defined geometry classes in the KBE module of the CAD software. These classes use predefined classes to create specific geometry. A number of parameters (e.g. dimensional) are used in the object definitions inside each of these classes. These parameters are used when generating the analysis models.

Figure 3 shows an example of how the KBE classes can be organized. The geometry classes are ideally organized in a hierarchy and contain rules needed to generate all geometrical objects of the system. The system contains of a number of subsystems that in turn contains a number of components. Each component is built up by a number of instantiations of part classes but also geometrical objects that are instantiated exclusively for each class. The part classes usually instantiate geometry that is used by several components.

The analysis classes can be of several types: 1) a complete analysis – i.e., a class that uses the geometry generated by the geometry class and analyze one or several properties, 2) pre-processing – using the CAD functionality to perform e.g. meshing, boundary conditions, or 3) creating an API call or Macro for later use during the analysis cycle. Some analysis classes can have children but these children still operate on system level. Some analysis classes start from the component level and use all component level results to create the system level results while other classes read parameters and other geometrical object data from the system level geometry class (the system level connection is indicated by the dashed lines).

The reason of not arranging all analysis classes under one main analysis class is that the API calls and Macros are used to instantiate the classes instead of having one main class that instantiate all analysis classes.

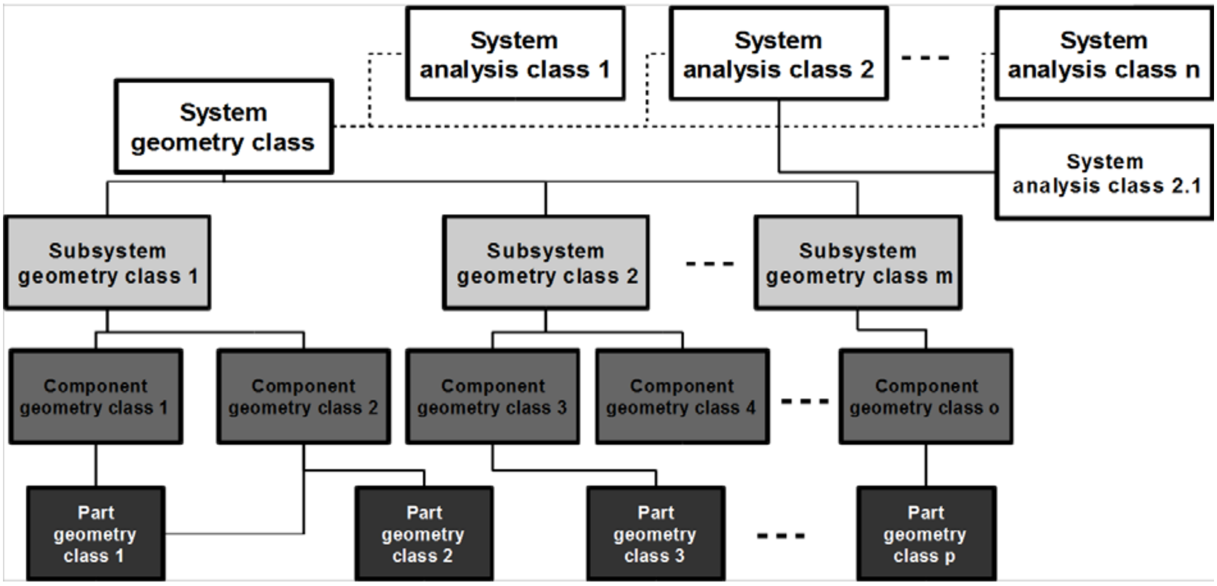

Figure 3. Example MM KBE class scheme

3.3 The API calls and Macros

The API and Macros are a part of the CAD software and aid the creation of the design and analysis loop where the KBE module of the CAD software needs to be complemented in functionality. Therefore, API calls and Macros can be used to instantiate KBE classes that help to automate the design and analysis loop. Macros can both be recordings of design and analysis activities as well as calls of CAD/CAE functions through the API. KBE classes can be used to name geometrical objects (e.g., faces, bodies) that are used by API calls later in the design and analysis loop. KBE classes can also be used to actually code (in e.g. Visual Basic, C++) some API calls by writing geometrical parameters (e.g., dimensions) into the code.

3.4 Generation of analysis model

Based on user input the KBMM approach can be used to generate or link a number of analysis models. The analysis model generation can be done in two ways: CAD-based and rule-based. The CAD-based method uses the actual CAD model generated by the KBE classes to create other models. The rule-based way uses the rules in the KBE classes to generate models. One example of the rule-based method is when an input file to a solver is written using geometrical parameters (or rules) that are defined within the KBE class. The file writing can be done using KBE functions to create, open and write files. An additional option to accomplish this is by external programs (e.g., MATLAB) called by a function within the KBE software.

Some models need input from other models. When all analysis activities are done the managing unit compares the results with the defined design objectives and constraints. If needed, a new iteration is initiated until an optimal design is found or a maximum number of iterations is reached.

4 JET ENGINE EXAMPLE

In the research project METOPIA (http://www.ltu.se/tfm/fpd/research/projects/METOPIA?l=en), a simplified, yet illustrative, turbo fan jet engine example demonstrates how the KBMM approach can be applied. Figure 4 presents an overview of the design and analysis loop that was used to find an optimal jet engine structure. It contains of six major activities; (1) *User interaction* (2) *Optimization*, (3) *Automated geometry generation, mass and manufacturing cost analysis*, (4) *Automated finite element model generation*, (5) *Automated rotordynamics analysis*, (6) *Automated displacement due to rotor-dynamics loads analysis*.

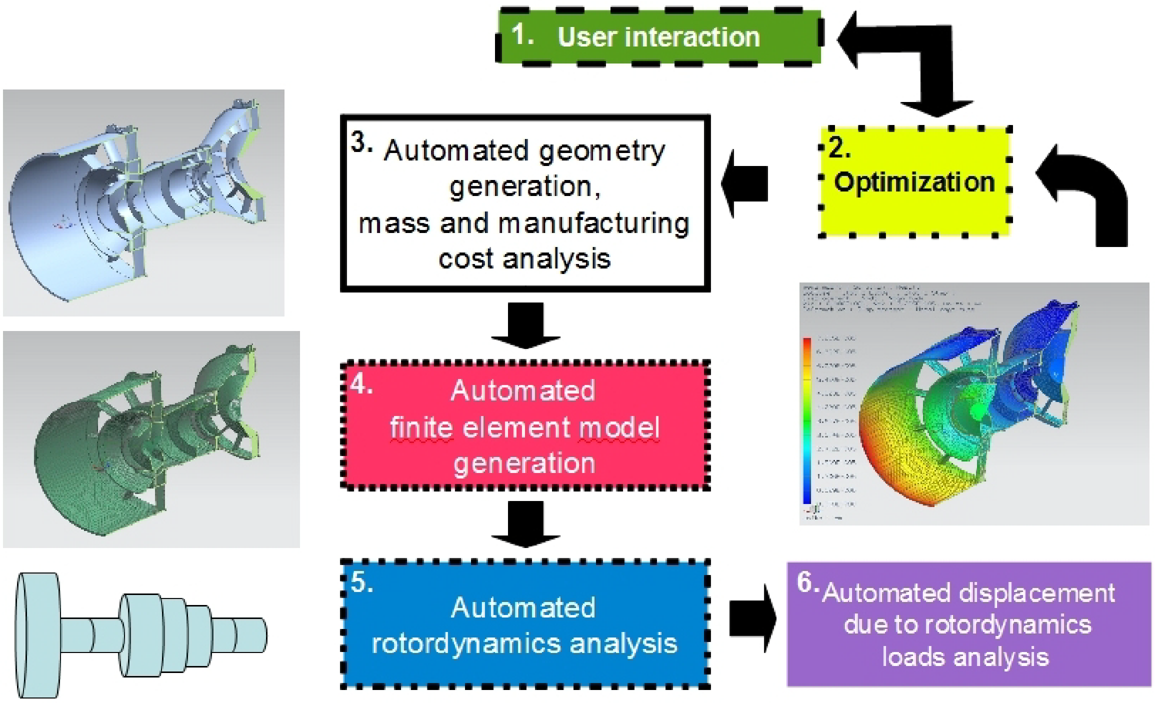

Figure 4. Design and analysis loop

4.1 Automated analysis activities

Further details of the automated loop are shown in Figure 5, where sub-activities and their corresponding software are presented. Activity 1 starts with the user choosing optimization variables, objectives and constraints and then starts the second activity: optimization. The second activity writes the input file that is used by the geometry generating KBE classes. The CAD-software NX is started where a Macro is run in the Gateway application to initiate a chain of journals (visual basic-based code that through the API executes NX functions), METOPIA actions, which are started from the user-defined menu and marks the start of Activity 3. In the NX Modeling application, all geometry is generated (i.e. fan case, low pressure compressor, intermediate compressor case, high pressure compressor, combustion chamber, high pressure turbine, turbine mid frame, low pressure turbine and turbine rear frame) and .dat-files containing mass and manufacturing cost analysis results are written. The file containing the mesh and material properties for the rotordynamics analysis is also generated (part of Activity 5). In Activity 4 the geometry is united to four bodies, each being one subsystem: 1) Fan case, 2) low pressure compressor and intermediate compressor case, 3) high pressure compressor, combustion chamber, high pressure turbine, turbine mid frame, low pressure turbine and 4) turbine rear frame. Note that subsystem 1 and 4 have only one component. A journal needed for the later coming add material activity is written. In the next stage the geometry faces and bodies are named to be used in e.g., mesh-mating and when material is added. NX Advanced Simulation is started and boundary conditions are added to the finite-element model and mesh-mating conditions are defined for the body interfaces. Nodes are generated for each bearing that will interact with the structure and these nodes are connected with 1D elements to the structure creating a so-called spider-mesh. Each body is 3D meshed with Tet4s, mesh size is adjusted automatically and separately for each body according to body size and mesh-mating conditions. Material is assigned for each body, then NX is switched to the .sim-file where requested output (e.g. displacements) from the finite-element analysis is assigned and input file (.dat) for NX Nastran is written, continuing Activity 5. This input file is edited by adding two lines of a code to punch out the stiffness matrix for the bearing position nodes. The earlier generated input file is then used to run the rotordynamics analysis in MATLAB. Activity 6: The resulting forces are added back to the finite-element model and NX Nastran solves for displacements. All NX files are closed as well as NX and results are analyzed. Based on the results a new input file is written for the next iteration.

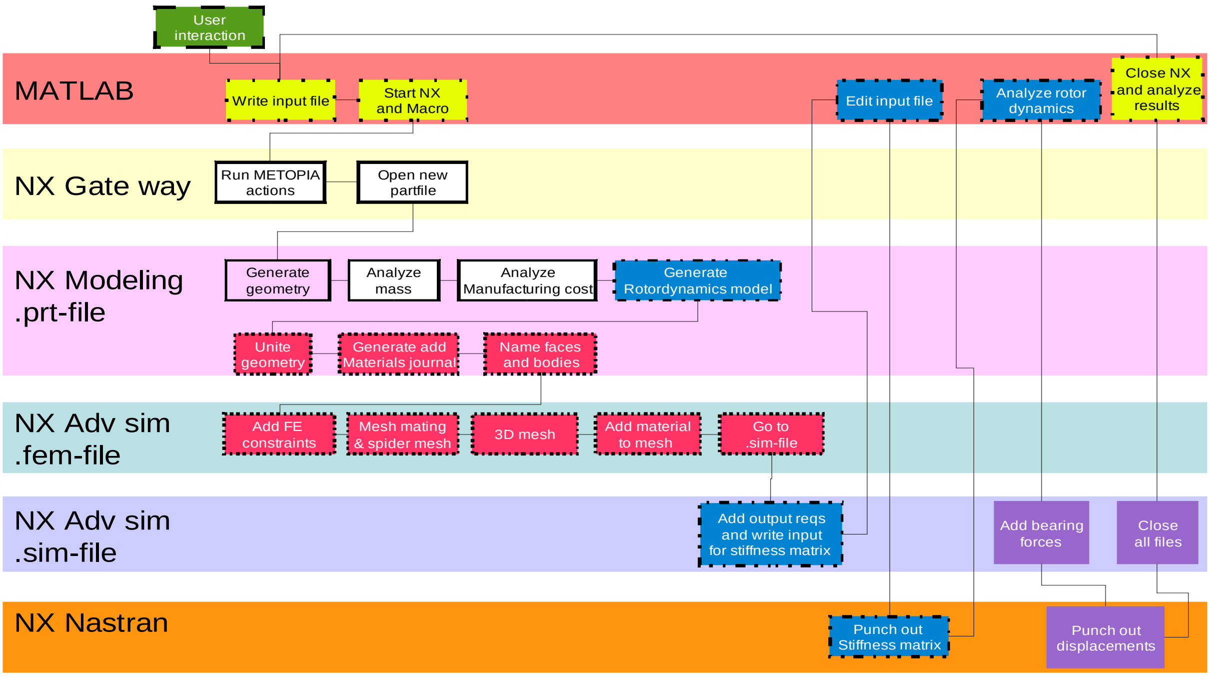

Figure 5. Design and analysis loop activity details

4.2 Analysis models

The analysis models used in the example include geometry, weight and manufacturing cost models as well as a finite element model and rotor model. Siemens NX 7.5 and Mathworks MATLAB 2009B were used to implement the models. All geometry and models are generated particularly for each iteration and no models are reused in the later operations.

The geometry is generated based on 18 user defined KBE classes and each jet engine component has its own class. The rest of the classes represent common geometry and are instantiated in several component classes. Common geometry includes: cases, flanges, struts and mount lugs. All geometry can be configured in different ways, e.g. changing lengths, radii, number of struts, thicknesses, cone angles (for intermediate compressor case, turbine rear frame).
As volume is easy to compute in CAD-models the weight can also be found. The following equation was used to compute the manufacturing cost ($Cost_{manufacturing}$):

$$Cost_{manufacturing} = m\, Cost_{material}\, k_{manufacturing} \quad (1)$$

where m (kg) is the mass, $Cost_{material}$ ($/kg) is defined by material choice (Titanium 6Al 4V, Titanium 6Al 2Mo 4V, Inconel 718, aluminium, steel) and its price per kilo, manufacturing method (cast, forged or fabricated) is also included into the manufacturing cost. For cast 15% more material is added and for forged and fabricated 10% and 5% were added respectively. The coefficient $k_{manufacturing}$ is used to tune the model to higher fidelity models where the manufacturing cost calculation is more elaborate.
The finite element model consists of Tet4 elements and has approximately 33000 nodes depending on the geometry configuration. The bodies with different material and element properties are connected together using mesh mating conditions. The rotor model is connected to the structure using 1D rigid elements. The finite-element model is regenerated at each iteration.
The rotor model consists of 9 cylindrical beam elements; each length and diameter is governed by the product definition and is an example of a rule-based analysis model generation. The stiffness matrix is created for the bearing nodes and used in the rotordynamics analysis governed by MATLAB.

4.3 Design study

The ultimate goal of developing analysis and simulation models is to have the ability to conduct design studies and evaluate what-if design scenarios. In this paper, a relatively simple design scenario was used to demonstrate the usefulness of the presented modeling approach. In particular, it was investigated how relatively limited design changes in component level impact system behavior. This seemingly limited design optimization study is in fact quite significant from a tier-1 supplier point of view: being able to evaluate such design scenarios rapidly increases the supplier's advantage against competitors and its ability to negotiate system-level design with the original equipment manufacturer. Specifically, it was investigated how the number of intermediate compressor case struts impacts mass (and thus cost, as the latter is a proportionate function of the former) and structural integrity when considering a fan-blade-off loading condition.
An optimization approach was used to investigate this design scenario. An optimization problem was formulated to minimize the mass of the jet engine structure subject to a maximum displacement constraint caused by the load generated by the imbalanced mass due to the lack of one fan blade (the actual impact of the blade on the engine is not considered). The design optimization variable was the number of struts for the intermediate compressor case. The initial guess was 12 struts while any number from 5 to 20 struts was considered. The displacements constraint was set to $0.5 \cdot 10^{-6}$ m.
Even though the design optimization variable was discrete, a gradient-based optimization algorithm was used to solve the optimization problem. Specifically, the MATLAB implementation of the sequential quadratic programming (SQP) algorithm was utilized. A gradient-based algorithm was possible to use because when considering a single integer optimization variable, the gradient computation can be manipulated to evaluate the neighbors of the optimization variable value (e.g., the values 7 and 9 of an incumbent iterate equal to 8), and thus guide the gradient-based algorithm to find a solution to a uni-variate discrete problem. Given the displacement constraint value mentioned above, the optimization converged to the lower bound of 5 struts. The displacement analysis results for this design are shown in Figure 6.
The next step of our research effort will be to formulate and conduct full-blown optimization studies. Obviously, the size and extent of such studies will depend on practical issues such as wall-clock time for one function evaluation (currently about 5 min) and the presence or not of mixed variable types (continuous and discrete). In the latter case, advanced derivative-free optimization algorithms based on mesh-adaptive direct search (MADS), such as NOMAD [14] will be used. Such algorithms may require hundreds of function evaluations for moderate-size problems (tens of variables), but are remarkably effective.

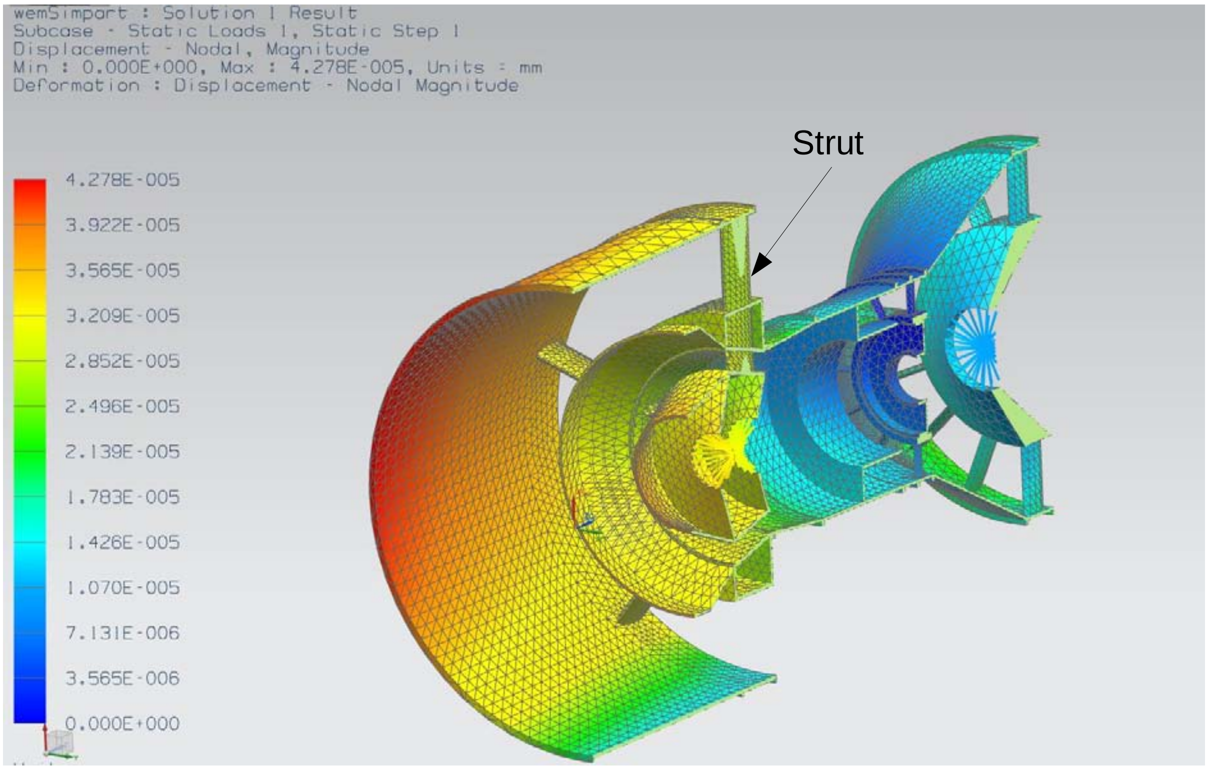

Figure 6. Displacement for 5 intermediate compressor case struts.

5 CONCLUDING REMARKS

A knowledge-based master model approach has been presented in this paper. In comparison to conventional master model approaches, e.g., [13], this approach is argued to be more flexible since the KBE classes can generate more radical geometry topology changes compared to traditional parametric models. In addition, the KBE classes can be reused since they are created in an object-oriented environment that promotes easy instantiation of the classes. In comparison to [3] this paper elaborates on the details of the constituents of the KBMM approach and presents new design optimization capabilities.

By using Macros and API calls to instantiate the KBE classes, the same Macros and API calls can be used even if the KBE classes are slightly updated. It was found to be beneficial to create smaller NX Journals that are NX session independent to maximize re-usability instead of grouping many functions into one Journal.

By having a top class that contains all optimization variables only one file needs to be edited by the managing unit at each iteration. The geometry classes was during this research effort updated for a new jet engine configuration, based on data from the EU FP7 integrated project: CRESCENDO (http://www.ltu.se/tfm/fpd/research/projects/crescendo?l=en), and most of the rules could be reused for the new configuration. Knowledge acquisition for the KBE model creation is not the focus of this paper; readers interested in such techniques are referred to [7].

When KBE software is part of CAD software the transition to detailed design is easier compared to dealing with discipline-specific models since detailed design is often based on CAD models.

Compared to using commercial software environments e.g. iSight, ModeFRONTIER, the KBMM approach is argued to be implementable in any CAD-software that has an API from where all CAD functions can be reached. Using MATLAB as the managing unit is argued to be beneficial since many organizations have MATLAB licenses. The geometry data export capability of the CAD platform determines which analysis models can be linked. Nevertheless, KBE software can be used to write input files to CAE software external to the CAD platform even though definition of mesh coordinates may need extensive coding of the KBMM.

The optimization will be developed further for larger and more elaborate design problems, and other optimization algorithms (e.g., effective and efficient derivative-free instead of gradient-based) will also be considered. Currently one iteration takes around 5 minutes using a 4GB RAM, dual core (2.8Ghz) processor computer. The example in Section 4 is relative simple, but demonstrate the concept for the KBMM approach and shows how whole jet engine design and analysis can be conducted, which is one of the objectives in CRESCENDO.

As the KBMM links analysis models to one product definition and enables optimization it is argued to be beneficial for early design and analysis. Automated model reconstruction implies potential time savings but also quality assurance since company-approved work practices can be used. As KBE is found most useful for routine design tasks the KBMM approach is believed to mitigate problems for standard engineering changes in designing complex systems such as jet engines.

ACKNOWLEDGMENTS

The financial support of VINNOVA NFFP5 (National Aeronautical Research Program, phase 5) is gratefully acknowledged. The financial and technical support of Volvo Aero Corporation is also gratefully acknowledged. Company-specific data of the jet engine model have been altered due to proprietary reasons.

REFERECES

[1] Kossiakoff, A. and Sweet, W.N., *Systems Engineering Principles and Practice*, (Wiley Series in Systems Engineering and Management, 2003).

[2] Newell, M.E. and Evans, D.C., Modeling by computer. in IFIP Working Conference on Computer-Aided Design Systems. Austin, Texas, USA, 1976.

[3] La Rocca, G. and van Tooren, M.J.L., Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach. *Journal of Design Research*, 2007, 5(3), pp.333-352.

[4] Sandberg, M., Kokkolaras, M., Aidanpää, J.-O., Isaksson, O. and Larsson, T., Whole jet engine analysis and design optimization: a master modeling approach, in *8th World Congress of Structural Multidisciplinary Optimization*. Lisbon, Portugal, June 2009.

[5] Dixon, J.R., Knowledge-Based Systems for Design, *Journal of Mechanical Design*, 1995, 117, pp11-16.

[6] LaCourse, D.E., *Handbook of Solid Modeling*, (McGraw-Hill, Inc, New York, 1995).

[7] Stokes, M., "*Managing Engineering Knowledge - MOKA: Methodology for Knowledge Based Engineering"*, (ASME Press, 2001).

[8] Chapman, C.B. and Pinfold, M., Design engineering - a need to rethink the solution using knowledge based engineering, *Knowledge-Based Systems*, 1999, 12, pp.257-267.

[9] Isaksson, O., A generative modeling approach to engineering design, In *International Conference on Engineering Design*, Stockholm, Sweden, 2003.

[10] Sandberg, M., Boart, P. and Larsson, T., Functional product life-cycle simulation model for cost estimation in conceptual design of jet engine components, *Concurrent Engineering: Research and Applications*, 2005, 13(4), pp.331-342.

[11] Lee, S.H., A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, *Computer Aided Design*, 2005, 37, pp.941-955.

[12] Hong-Seok, P. and Xuan-Phuong, D., Structural optimization based on CAD-CAE integration and metamodeling techniques, *Computer Aided Design*, 2010, 42, pp.889-902.

[13] Hoffmann, M.C. and Joan-Arinyo, R., CAD and the product master model, *Computer-Aided Design*, 1998, 30(11), pp.905-918.

[14] Audet, C., and Dennis, Jr., J. E., "Mesh adaptive direct search algorithms for constrained optimization," SIAM Journal on Optimization, 2006, 17(2), pp.188–217.

Contact: Marcus Sandberg
Luleå University of Technology
Department of Business Administration, Technology and Social Sciences
SE-971 87 Luleå
Sweden
Tel: Int +46 920 493072
Fax: Int +46 920 491399
Email: marsan@ltu.se
URL: http://www.ltu.se/?l=en

Dr. Sandberg is currently an Assistant Senior Lecturer at the Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Sweden. His research interests are in the areas of knowledge-based engineering, systems engineering and simulation-driven design.

Dr. Tyapin is a Post Doc at the Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Sweden. He has among others done research considering optimization of industrial robots such as Parallel Kinematic Machines in collaboration with ABB Robotics, Sweden. In 2009 Ilya has completed his PhD in Mechatronics and Computer Science at University of Queensland, Australia.

Dr. Kokkolaras is a Visiting Professor at the Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Sweden. His primary appointment is Associate Research Scientist at the Department of Mechanical Engineering at the University of Michigan, in Ann Arbor, Michigan, USA. His research interests include multidisciplinary design optimization of complex engineering systems, uncertainty quantification, platform-based design of product families.

Prof. Isaksson holds a position as senior company specialist in Engineering Design at Volvo Aero in Trollhättan, Sweden. He has an MSc in Mechanical Engineering and a PhD in mechanical engineering, both from Luleå University of Technology. He joined Volvo Aero in 1994 and is currently responsible for technology and methods development in the area of engineering and product development.