Development of Design and Engineering Engines to Support Multidisciplinary Design and Analysis of Aircraft

Gianfranco La Rocca, Michel J.L. van Tooren
Design, Integration and Operations of Aircraft and Rotorcraft (DAR)
Faculty of Aerospace Engineering
Delft University of Technology
G.LaRocca@lt.tudelft.nl

1. Introduction

In 2002, NASA and the Advisory Council for Aeronautics Research in Europe produced two relevant documents, the Aeronautic Blueprint and the Strategic Research Agenda, where the first century of aviation is briefly analysed and the technological challenges for the next 20 years are set. In a couple of decades the aeronautical systems will differ from today’s systems at least as much as the actual systems differ from those of 1930. The aeronautic community will have to face such a challenge in a problematic socio-economic scenario, where the availability of economical and intellectual resources is shrinking upfront the increasing complexity of demanded products. A fundamental paradigm shift is required to pass to a new knowledge based vision of business, where knowledge needs to be engineered and managed as a key business asset. Knowledge Management (KM) and Knowledge Based Engineering (KBE) represent two organizational and technical disciplines that can support this knowledge paradigm shift and help companies to retain their competitive advantage in the engineering market. KM can provide the vision and the strategy to make the best use of the people involved in products development, and freeing more time for creativity and innovation at expense of repetitive and non-adding value activities. Knowledge Based Engineering can provide a technological solution to implement this vision in the field of design and engineering. A critical analysis of designers needs during the evolutionary phases of the aircraft design process reveals the need of more advance tools able to support the conceptual stage of design, as well as enable a smooth transition to the following design phases, without creating discontinuities in the knowledge generation flow.

In this paper the KBE design approach is introduced and its impact on the design process is discussed in perspective with the traditional design methodology. This high potential design technology has been exploited to develop a complex knowledge based modeling environment, the Multi Model Generator (MMG), able to support designer in the conceptual and preliminary phase of aircraft design. The development of a set of high level primitives (HLPs), which constitute the fundamental components of the MMG, is discussed with particular emphasis on their capability to capture the view of designers on the aircraft product and automate part of the design and analysis process. The paradigm of a new modular design and engineering environment, referred as the Design and Engineering Engine (DEE), is introduced as a viable solution to attack complex multidisciplinary problems, exploiting engineering skills and analysis tools from many different design experts, eventually dispersed outside the boundary of the single company walls. In particular, the role of the MMG within the DEE and its key role to provide customized models for the different disciplines tools connected in the federated design and analysis environment are discussed. A prototype of the software framework now under development, to link the various DEE components and control the design process activities, is described as introduction into a new research direction.
2. The impact of knowledge based engineering on the design process

In the design process of a product, such as an aircraft, a car, or a generic mechanical component for example, a diverging and a converging phase can be generally distinguished (La Rocca, Krakers, van Tooren 2002; van Tooren 2003). During the first conceptual phase, many potential solutions are synthesized to comply with the list of requirements provided by the market/customer. The broader is the amount of proposed solutions, the higher the chance to have enclosed the most appropriate or the closest to the best. These solutions can be either variants of one product concept, or completely different product configurations, which, in any case, must be analysed and eventually optimized in order to perform a fair trade-off. This will initiate the converging phase of the design process, where the best solutions are selected for the next design level. This diverging-converging process is actually strongly iterative and requires a continuous adaptation and modification of each proposed configuration during the design loops. Large amount of data and information are continuously generated and exchanged across the various involved disciplines operative with a large suite of design and analysis tools. Many different models need to be (re)generated and adapted for the different process stakeholders. Generated outputs from one design and analysis tool often need to be re-processed in order to be transformed in usable inputs to others.

As illustrated by the design cube, in Figure 1, a thoroughly exploration of the 3D design space requires the capability to move along the three directions. There are not only the typical time-defined conceptual, preliminary and detailed design phases, but the product has to be continuously examined from the point of view of the various discipline stakeholders. The capability to change the focus on the product needs also to be supported, which implies the capability to get simplified models of the global aircraft as well as detailed models of isolated subsystems or components. An exhaustive exploration of this multidimensional space calls for an organised and integrated involvement of many different experts, together with many analysis tools, which need to be agilely plugged-in and out during the evolutionary design process.

![Figure 1: The design cube. Multi-phase, Multi-disciplinary, multi-scale.](image-url)

The traditional design approach shows some inherent limitation in handling such complexity with efficiency and effectiveness. The various models required by the discipline involved in the design and analysis process are generally very different by nature, because they have to reflect the very different views that different discipline expert have on the same product. Since the design/drafting and analysis stages are generally carried by different persons using different software tools, the whole design scenario results populated by a lot of non-synchronized non-relational models, leading to a high risk of analysis inconsistencies. Due to the large extent of
manual processing and data acquisition, the set-up of all these models results very time expensive: a critical bottleneck for the whole design process. Whenever major changes are required in the product configuration, because dictated by the need to improve the product behavior or add extra functionalities, all the models available at the moment become immediately obsolete, as well as many of the analysis results. New models that reflect the design change must be generated and analysis performed again. The experts get frustrated by all the time consuming and repetitive activities, without considering the fact that, for each manual iteration there are possibilities to generate new errors. As a result, the inertia of the design process is such that less design iterations can be performed than those actually necessary, less what-ifs can be investigated, and many possible product configurations and variants must be discarded a priori. Mature in-house knowledge very often prevails over unproven and risky innovation, which, in many cases, leads to the elimination of promising ideas before any fair assessment. Quality and innovation are the first victims; a real multidisciplinary design is almost impractical.

Figure 2 (left): The Knowledge-Based engineering design process. Figure 3 (right): The product (or generative) model.

The schema of Figure 2 represents a possible design paradigm through the implementation of KBE. The main difference, respect to the traditional design schema discussed above, consists in the new pivotal role assigned to the product model, which represents the central repository of the design knowledge within the process. This implies that the relevant knowledge of the design team and of the experts involved in the various disciplinary analysis activities has first to be captured and then opportunely translated into that set of engineering rules that actually makes up the product model. The product model is actually a computerized application that KBE developers code using a high level programming language, which has been enriched with sets of commands and methods to drive an integrated parametric CAD kernel. This integrated environment where Artificial Intelligence meets CAD represents the most relevant feature of KBE systems. The functionality of the product model can be described by the simplified representation of Figure 3 a set of input values is provided to the parameters used in the rules-base, the KBE system applies the rules which process the input values and the engineered design is automatically generated as...
output (Cooper, Fan and Li 2001). With little or no human interaction, the product-model is able to generate geometry, or some other models, eventually not including any geometrical entity. The designer actively interacts with the design process through the editing of the product model input values. The bottleneck at the interface between the design/drafting and the analysis phase can be resolved by capturing in the product model the rules to generate automatically the various discipline specific models. Separate but fully relational models can be generated to guarantee the consistency of analysis results. The benefits of this integrated and largely automated approach are huge: since the expert is not asked anymore to manually assemble new analysis models when changes occur in the product configuration, the design process results accelerated and many more iterations can be performed in the same time normally required by a single manual iteration. The designer should feel the confidence that everything can be easily changed; the impact of every design choice on the final properties of the output product can be faster evaluated and eventually withdrawn.

3. The Aircraft Design Process and Designers’ Needs

Among the available tools nowadays employed to support designers during conceptual design of aircraft, CAD systems are far the most common and widely used. Current generation CAD systems are mainly feature-based, which means that they have a standard set of parameterized primitives (points, lines, solid volumes, holes, chamfers etc.) that can be adjusted and combined together to represent a design. As a matter of fact, CAD primitives have very poor knowledge recording and learning capabilities, thus they provide conceptual designer a rather inadequate mean to support the knowledge work and the level of design abstraction typical for this early phase of the design process. For a CAD program an aircraft wing will always be a set of surfaces and solids, never, for instance, a lift generating object compiled of different wing sections with leading and trailing edge devices and an internal generic structural concept. However, if we consider the way a conceptual designer thinks and the way he looks at design problem, we find that actually it is this global modeling approach that is required. What the designer needs is an efficient way to virtually manipulate ideas and create many different design solutions. Then an efficient way is needed in order to assess and compare these possible solutions in a fair trade-off. Hence designers should not be hampered by the manipulation of infinite sets of lines and points, but provided with a limited set of high level primitives, easy to adjust and link together, with inherent functionalities and knowledge capability (that is the capability to autonomously act and react to the occurrence of certain events). These primitives should be able to fulfill functionalities such as generate lift, provide control and thrust, carry loads, store payload et cetera (van Tooren et al. 2003). Furthermore they should be able also to automate (part of) the activities required for the aircraft assessment, such as the generation of models for structural and aerodynamic analysis, or cost estimation, or tooling for manufacturability study and so forth.

A solution is needed to gain more knowledge about the product already in the early stage of the design process, which implies the need to fill the gaps and discontinuities between the various design phases (namely conceptual, preliminary and detail phases). If a design tool is developed to target just one specific phase of the design process, discontinuities in the flow of data, information generated during design might occur. For example geometry models generated by purely conceptual design tools are typically suitable just and only for visualization purposes. When advanced analysis tools such as CFD or FEM are required for a proper assessment of the concept, new models with the appropriate level of refinement have to be generated from scratch, in separated design environments. It should be considered that nowadays in order to handle the
complexity of the design process, aircraft configurations are generated largely based on semi empirical models and statistc methods rather than first-principle. About the 80% of the total life cycle costs of the final product turns out to be determined by the use of low fidelity models (Staubach 2003). The reliability of the obtained results is often arguable and, as soon as new and non-conventional aircraft concepts need to be evaluated such methods result inadequate, because of the lack of any previous reference and statistical data.

On the other side, there are continuous attempts by industry and academia to develop comprehensive integrated design tools able to handle all the modeling, analysis and optimisation phases, at a relatively high level of detail. This ambitious approach has often led to an irresolvable level of integration between design and analysis components, with significant drawbacks concerning the possibility to further develop and maintain such complex system. Furthermore the exploitation of such complex tools within broader collaborative design environments results always very problematic. Experience has shown the difficulty to substitute one of the analysis functionality integrated in such design system with a different analysis tool provided by another party; or to use only some of the tool functionalities, spilling data streams directly from the internal flow or by-pass some modules.

4. **Use of KBE Technology to Define New High Level Primitives**

Given a set of top level requirements, a good designer would like to investigate more aircraft concepts and see which one has the best potential to fulfill them, for example a traditional airliner and a blended wing body aircraft configuration might be good candidates. These apparently very different design solutions are actually linked by several elements of similarity: they all feature some components, with the common purpose to fulfill certain functions (generate lift, supply thrust, provide control and stability, accommodate payload et cetera). Even if these components have different shapes and are recombined in different topological configurations, an object oriented analysis of the aircraft model shows that is possible to abstract a certain amount of classes, which can be instantiated in wing, fuselage, engines and connection objects (see Figure 4). These classes of objects, able to capture the similarity element linking the different aircraft components, are actually what we addressed in the previous section as the High Level Primitives (HLPs): generic entities with a similar functionality, shape and behavior.
The HLPs can be interpreted as special bricks, kind of rubber LEGO blocks, which can be individually morphed thanks to their full parametric definition, and combined together in order to generate a potentially infinite amount of different aircraft configurations. KBE, which fully supports object-oriented modeling, has been used to develop a computerized modeling environment where the conceptual designer can define and combine these primitives to agilely build a virtual representation of the aircraft concept he has in mind. In Figure 5 it is shown that, just using four of these HLPs, namely the wing-trunk, fuselage-trunk, engine parts and connection element primitives (La Rocca and van Tooren 2002a, Koopmans 2005), it is possible to generate a broad range of aircraft configurations and almost infinite variants of each configurations.

The flexibility of the HLP, here intended as the capability to represent objects within a large range of typicality, is directly related to the choice (and number of course) of parameters used to define the corresponding class. In order to design a wing, for example, the designer can use multiple instantiations of the wing-trunk and connection HLPs, and define the shape and topology of the wing, just adjusting the values of parameters such as span, chord length, sweep and twist angles, number, location and type of airfoils and many others.

The definition of the HLPs is not just limited to the parametric description of aerodynamic surfaces, but includes also some advanced generative capabilities to model internal structure configurations. Through a specific set of parameters, the designer has the possibility to modify the position of the internal structural elements (ribs, spars etc.) as well as to change the topology of the structure configuration (i.e. change the number of spars, ribs etc.). Since the definition of the internal structure is associated to the shape of the outer surface, it will automatically adapt when the latter is modified for example by a different wing design (La Rocca and van Tooren 2002b).

This capability to strongly affect the topology of a model configuration sets a big difference between classic parametric CAD and KBE HLPs based models. Indeed the HLPs definition via a high level programming language offers the possibility “to teach” the HLPs how autonomously handle constraints during any topological variation. Furthermore the ruled-based approach used to define the primitives offers also the possibility to handle some known native CAD kernel limitations and bugs. Typically, when an experienced CAD operator realizes that a given geometric operation often fails he knows how to implement a different approach to work around the problem. This very approach can be directly programmed in the body of the HLPs, such that workaround procedures can be automatically triggered, either proactively or after a failed operation has been diagnosed.

There is another fundamental element of similarity that links the design of very different aircraft configurations, such as the two of Figure 4, the KBE technique can capture and harness through the HLPs definition. Indeed, the assessment process of very different aircraft configurations calls for very similar analysis methods and relative procedures. KBE can be used to capture and manage also process knowledge, not only the what, but the how as well. For example specific models will have to be generated for CFD analysis in order to evaluate the aerodynamic characteristics, or for FEM analysis to validate the soundness of the structural design, or for weight and balance module to assess the stability margin and so on (more details are provided in next section). A very large deal of these preprocessing procedures consists of repetitive activities that can be captured and formalized into explicit rules, such that the various HLPs can reapply them systematically and autonomously. In this way the designer can be relieved from the burden of time consuming routine activities and the conditions for extensive engineering automation are created. This represents a valuable use of Artificial Intelligence in aircraft design, because
addressed at capturing and automating the non creative part of designers’ work, where designers actually need the most support, rather than in the creative process of thinking about a feasible aircraft concept.

5. **A flexible multi model generator to support aircraft multidisciplinary analysis**

As anticipated in the previous sections, in order to perform a multidisciplinary design of aircraft, a number of sub models have to be generated for the disciplines involved in the analysis and optimization processes. These sub-models represent the specific viewpoints the various design process stakeholders have on the aircraft. For example the aerodynamicist’s view on the aircraft mainly consists of the outer aerodynamic surfaces, without any direct interest in the internal structure configuration, neither in the location of the various aircraft systems. By the way, the consistency of these sub-models is as much important as their tailoring. For example, if the wing planform design is changed, updated structural models suitable for stress analysis and wet-surface models for aero analysis needs to be promptly generated. If the aerodynamicist changes the thickness of the wing, such modification must be reflected also in the model of the structure used by the stress engineers, or in the mould line model used for tooling design. These two requirements call for the development of advanced and smart modeling tools, which are able to discern and distill the various disciplinary aspects of a product, and deliver them as sets of separate but fully relational sub-models.

In the development of an effective multidisciplinary design and analysis strategy, there are other important issues, apparently of non technical nature, which in the end turn out to have a significant impact on technological implementations. It has been proven, how crucial is the capability to give discipline experts the possibility to participate in the design process with their own analysis instruments (Morris 2002; Morris et al. 2004). The experts should not only be welcomed with their tools, but also put in condition to exploit them at best. This imposes demands both on the format to be used for data exchange between analysis and modeling tools, and on the level of maturity of the data exchanged, that relates to the level of pre-processing required on the exchanged data and information, to make them directly “edible” to the recipient analysis tools.

On the base of the issues discussed so far, a highly flexible and modular aircraft modeling tool, based on the High Level Primitives concept, has been developed with the ICAD KBE system and qualified as the Multi Model Generator (MMG). In synthesis the MMG defines an aircraft super class that, on the base of a large set of input parameters, can be instantiated into a specific aircraft design, which is again the result of the automatic multiple instantiations of other component-classes, namely the HLPs. Methods and rules are programmed in the body of the generic aircraft product model, such that for different input values, different amount and type of HLPs are selected, instantiated and assembled together to shape the desired aircraft configuration. As discussed above, the various HLPs contain the knowledge about how to generate their shape and perform some other tasks. When multiple HLPs are instantiated together to generate a complete aircraft model, the latter inherits as a whole the characteristics and the capability of the single HLPs/components. For example, if a given wing-trunk primitive is able to generate from its outer surface a specific model for a given aerodynamic analysis tool, then the whole wing (built up as an assembly of several wing-trunk primitives) will be able to generate from its complete surface a model for the abovementioned analysis.
6. **Structure and functionality of the multi model generator**

Structure and functionality of the MMG can be discussed referring at the oversimplified representation of Figure 6. Two main functional blocks can be pointed out, namely:

1. The product model, which is the main body of the MMG and contains the definition of the various HLPs and the rules to bring them together to build-up different aircraft configurations.

2. The reports writer, which is the set of utilities in charge to extract from the whole aircraft product model the data and information needed to build the specific sub-models (called reports in the technical implementation of the MMG), for the various analysis tools.

Prior to launch the MMG, the user has to fill the input file containing the list of all the parameters used to define the aircraft super class instantiation. The MMG is connected to an amount of external libraries and data repositories (containing airfoil data, fuselage cross section definition et cetera), whose contents are automatically retrieved when required for a specific instantiation. The user, still via the input file, can indicate which specific reports he is interested to get generated by MMG. The KBE platform used to define the MMG support an interesting feature, called *Demand-driven instantiation*, which has a large impact on the efficiency and computational time required for model computation: when the generation of specific reports is requested to the MMG, not the entire aircraft product model is computed, but just the branches which are strictly required to match the specific user request. For example, when a user asks for the generation of the aircraft wet-surfaces report, the MMG will not perform any operation concerning the generation of internal structure, or the distribution of the various aircraft systems and so on.

As shown in Figure 6, the reports-writer block represents the actual link between the modeling environment and the analysis disciplines. It is here that the processing capabilities coded inside the HLPs are invoked and data and information are generated and collected in such a way to create input models for specific analysis tools. In particular the links between the MMG
and the structure and aerodynamic disciplines are discussed, as examples of a methodology to integrate the KBE modeling engine with external analysis services, without any attempt to duplicate or incorporate any of those analysis capabilities directly inside the MMG. The way the MMG has been developed guarantees several advantages:

- The MMG can be operated interactively, editing the input file via keyboard and evaluating the changes in aircraft model directly in the MMG graphical browser.
- The MMG can be operated remotely. A complete input file can be edited by a remote user (N.B. this operation can be also automatically performed by an optimization tool, when the MMG is operative within an interconnected analysis and optimization environment) and then submitted to the MMG for a *batch run*. The output reports can be retrieved via web, after the MMG has signaled his ready status.
- As a consequence of the above, many *non-geographically collocated* users can use the MMG submitting their customized version of the input file and their list of required reports.
- For a given version of the product model (stored with an appropriate file revision system), each input file defines univocally one aircraft configuration/variant. It might be very efficient to store just one copy of the given input file version (and re-submit it to the MMG whenever required) rather than store a multitude of large output models/reports, generated with that given input file.
- The MMG code, even if featuring several hundred of thousands of code lines, has been developed with a modular approach. Each HLP for example is a module, which again features a surface-generator-module and a structure-generator-model. Again these modules are composed by various “capability modules” (such as split-in meshable-surfaces and create-cloud-of-points described in the next subsections) of which some are shared by different HLPs. In this way maintenance and further development of the application is facilitated. When a single module is improved it can be simply plugged-in to replace the old version, provided that the *interface* remains the same. Relaying on a correct use of the interfaces system, new modules can be generated, to create new report writers for example, and bolted onto others to add extra functionalities to the MMG.

**Link with Structural Analysis**

Efforts have been made to create a seamless link between the geometry modelling and the FEM analysis environment, via a complete automation of the pre-processing and analysis phases, hence passing from the MMG native geometry definition directly to the post-processing phase of the FEM analysis results. The way this process is typically performed requires combined efforts and good communication between draftsmen and FEM experts. A lot of manual, lengthy and repetitive operations are required to assemble a FE model, starting from a CAD geometry model. One of the most time consuming, accordingly with the complexity of the geometry model, concerns splitting the model surfaces in *meshable* surface elements, that is surfaces with adequate aspect ratio and skeweness, no more than four edges, each edge matching with just one edge of the neighbor surfaces (La Rocca and van Tooren 2002b). Several interviews and other contrive knowledge capturing techniques (Stokes 2001; Shreiber et al. 2000) have been used to capture into sets of explicit rules and logical mechanisms the modus operandi of FEM experts in different situations with diverse levels of geometry complexity. These rules and mechanism have been implemented in the HLPs (in the capability module mentioned above) in such a way, every time a new input model for FEM analysis is required, the surfaces cutting routine can automatically take action and create as report a consistent set of meshable surfaces, no matter
how the current aircraft configuration might have been changed in shape and topology. The IGES format is used to transfer this purely geometry information, whereas a complementary data stream has been activated to bring outside the MMG the other meta-data required to automate the generation of FEM models (see Figure 7). A set of XML files (addressed here as FEM-tables) is automatically generated by the MMG, in parallel with the IGES files, to transfer all the relevant (non-geometric) information generated by the MMG. A smart PATRAN session file has been programmed in the PATRAN Command Language to complete the automated generation process of the FEM model and finally run a structural analysis and/or optimization, using one of the supported solvers.

![Figure 8 (left): Two structural models. Figure 9 (right): Generation of consistent models.](image)

Other report writers have been generated to link the MMG to other kind of structural analysis tools requiring for example a lumped masses and beams representation of the aircraft (see Figure 8 on the left from Koopmans (2005) or a solid elements discretization of the entire structure to exploit advanced FE formulation based on solid p-elements. See Figure 9 on the right from Lisandrino and van Tooren (2004). This diversity of model demonstrates how different the same product can appear to the eye of different experts, operating with different tools, in different location in the design space.

**Link with Aerodynamic Analysis**

Similarly to the structural analysis case, the goal here was to create a seamless integration between the MMG modeling environment and external aerodynamic analysis tools, either commercial off the shelf (COTS) and proprietary codes. The easiest form of link is based on the direct exchange of the MMG native geometry via IGES or STEP files. Eventually it is possible to preprocess the surfaces in order to match some specific requirements from the aerodynamic customer, such as split the surfaces in more patches and so on. This approach actually is feasible whenever the analysis tool to be linked with the MMG is able to accept standard file format such STEP or IGES. For other cases a special capability module has been generated in the HLPs to “translate” the aircraft surfaces into a cloud of points (see La Rocca and van Tooren 2002c), whose Cartesian coordinates are automatically evaluated by the MMG with respect to a reference datum and then written into an ASCII file, formatted to be readable by the given aerodynamic tool. This strategy has offered the possibility to deliver high quality surface model to powerful in house developed aerodynamic tools which were not provided with own adequate modeling capability. The MMG user has the possibility to adjust the amount and distribution of these points via a specific set of control parameter in the input file, to affect also locally the density of the cloud of points. Thanks to this flexible approach the points stored in the cloud can be used directly as mesh seeds for a grid (Qin et al. 2002), or as corner points for a panel discretization, or just as constraints to re-
spline a surface (Laban et al. 2002). Some functionalities have been included in the MMG to generate other kind of simplified aircraft model, and then write directly compile sets of NASTRAN cards for the vortex lattices model solution (Koopmans 2005). A high level of automation for the modeling-analyzing process has been reached also for high fidelity CFD analysis, where the MMG has been used to create an accurate 3D-aircraft model, inclusive of the surrounding computation control volumes and grid information. In the specific case FLUENT was the selected CFD package, and its preprocessor GAMBIT, driven by a case-specific journal-file automatically generated by the MMG, could perform all the needed preprocessing activities and launch the computation run (Sterkman 2002).

As indicated by the list of references mentioned here, the development of the High Level primitives and their capability modules is a continuous process at the DAR group of the Technical University of Delft. The object oriented definition of the HLPs allow an extensive reuse of coding and software modules in such a way to re-apply and the knowledge gained in one project to the specific need of new study cases.

7. Development of design and engineering engines

In this section the paradigm of a design and analysis environment, addressed as Design and Engineering Engine (DEE), is discussed to show the ideal framework where the capability of the MMG can be exploited at best. The basic structure of a DEE is shown in Figure 10: it consists of a set of properly interconnected toolboxes, which are either COTS or proprietary software modules, such that, as far as possible, automated multi-disciplinary design, analysis and optimization becomes feasible. The MMG constitutes the core unit of the DEE. The bi-directional interfaces system, which links the MMG with the various toolboxes, as well as the Initiator and the Converger modules form the other key components in the structure of the DEE. The Initiator box is in charge to generate the initial values for the parameters contained in the MMG input file. It should be noted that Initiator might consists of a computational module, eventually including some optimization capability, as well as another KBE module which are separated from the MMG. The Evaluator/Converger boxes provide the functionality to judge if the set of input generated for the MMG yields to an aircraft whose performances match the initial requirements, and if the convergence of a FEM analysis is reached on the base of forced changes in the mesh configuration.

Figure 10 (left): Design and Engineering paradigm. Figure 11 (right): Role of the MMG in the MOB project design engine.
In order to participate to DEE the various toolboxes should all be able to operate as stand-alone modules, and be accessible via a clear input/output interface. In this way they can be easily plugged in and out of the DEE, keeping high the level of flexibility and maintainability of the whole design and analysis system. According to an object-oriented (O-O) perspective on the DEE, the various toolboxes can be interpreted as sets of different methods that can be applied to perform conceptual and preliminary aircraft design. A successful prototype version of a DEE was represented by the outcome of the EC sponsored project MOB (Multidisciplinary Design and Optimization of Blended-Wing-Body Aircraft Configurations). In that project it has been shown how KBE can impact the design process and turn MDO from a great potential to a real working business. In Figure 11 the position of the MMG within the MOB DEE is represented. The MMG, starting from a unique definition of a BWB aircraft configuration, was in charge to extract a set of different, yet coherent sub-models tailored to the various analysis tools provided by a broad group of partners from the industry and academic world: low and high fidelity models for aerodynamic analysis, 2-D planform models inclusive of movables surfaces for aeroelastic analysis, structure models for FEM analysis and optimization, fuel tanks and systems masses distribution for weight and balance assessment. The MMG provided also the capability to focus on a specific detail of the aircraft, a door cutout in this case, and provide the base to apply a multi-level analysis and optimization strategy.

In van Tooren (2004) other examples of DEEs developed at the DAR group of the Technical University of Delft are discussed. The experienced gained so far has shown how this approach can be effectively used for many different study cases, ranging from the investigation of the ground effect of sport cars (Adriansen 2004), to the design of fuselage panels including piezo-elements for active sound damping (Krakers et al. 2003); from loads calculations for large commercial airliner (Cerulli, Mejer and van Tooren 2004), to manufacturing studies and costs estimation of aircraft components (van der Laan 2004).

The experience gained at DAR has shown that, every time the design process of a product involves an amount of different stakeholders, in terms of different experts and different design and analysis tools which must interconnect and exchange data and information, the set up of a customized DEE gives not only the possibility to speed up the process cutting the time wasted by repetitive activity, but provides the possibility as well to introduce and test new technologies, e.g. advanced analytical FE formulation, implementation of smart materials etc.

8. Towards an implementation of a DEE software framework

We have discussed in this paper how the use of KBE techniques can be used to automate large parts of designers work, capturing best practice and routine activities in computerized application. Tools such as the MMG (and the PATRAN and GAMBIT session files briefly introduced above) represent anyway islands of automation, which need to be connected within a smart framework able to create physical links between the various DEE modules, manage the exchange of data and information and control the execution of the various design and analysis tools. The DEE framework, in principle, must be able to let a multi-disciplinary design team operate with a heterogeneous set of design and computational tools running on different computers, mounting different operating systems and belonging to separate networks, eventually geographically not collocated.

An agent-based client/server architecture is now under development at DAR to provide a software framework for the DEE (see Figure 12). The core of the framework has been implemented in the high-level scripting language Python, because of its platform independency; its capability to
supports object-oriented programming and the large set of available library modules. As illustrated in Figure 12, the main components of the system are the DEE server and the various disciplines involved in the design process (represented in this case by their specific tools), which represent the DEE clients.

Every DEE client is embedded in a wrapper, which provides the access interface each tool needs to wear, in order to be allowed joining the DEE environment. The role of this interface is fundamental in view of the flexibility level we want to reach in the DEE assembly process. In order to create a real plug-and-play design and analysis environment, it is essential that each tool can be easily substituted by a new one (or an upgraded version of the same tool), and that is achievable only if the tool interface is kept consistent. This wrapper actually consists in an XML configuration file which contains information about how to trigger the tool, about the list of required inputs to operate the tool and the list of data (or models) to be expected as output.

Server and clients exchange administrative data: any DEE client that participates in the DEE must register to the DEE server, which in return provides the client with the list of all the other registered clients in the DEE and the kind of services they provide. Typical administrative data are hostname, IP-address, port-number for communication and others. The DEE server is responsible for checking that all clients in the list of registered clients are running and accepting connections. Once a client is registered, it is allowed to send a request (to produce some kind of data or model) to any of the other clients in the list of the registered clients. If the request can be satisfied by the requested client, a peer-to-peer connection between the two clients will be instantiated; the required data or model (the product data) will be generated and their storage location will be communicate to the requesting client in the form of a Uniform Resource Locator (URL). This URL is then used to retrieve the requested product data.

An advantage offered by this approach is that every client in the DEE is actually able to initiate the design process, because it is informed by the DEE server about which of the available clients in the DEE is able to provide the required inputs and how to connect it. The design process turns out highly dynamic and able to self-configure, without the need for a static, predefined process flow, as typical with most of the commercially available process integration software. The example described below will better illustrate the concept.
Example of a design scenario: structural analysis of a wing

In the diagram of Figure 13, the various steps of an automated process for the FE structural analysis of a wing are represented, to show how such a process can be automated through the implementation of the DEE framework described above. In Step 1, the structure analysis tools registers to the DEE server. The DEE server returns a list of all available DEE clients (in this case the MMG and aerodynamic tool), their address and the services they provide. Once the registration has finished, clients are allowed to have peer-to-peer connections. In Step 2 the structures client connects to the MMG and sends messages to request the wing structural topology and the meta-data to generate the FEM model. In Step 3, the structure client also sends a request to the aerodynamics client for the aerodynamic pressure distribution on the wing, such to generate a load case for the structural analysis.

At this point the aerodynamic tool needs a model of the wing surface to compute the pressure distribution, hence in Step 4 an implicit request for aerodynamic topology is sent to the MMG. When the MMG and the aerodynamic tool have satisfied the requests of the structure client, the latter is informed where it can retrieve topology, meta-data and pressure distribution, as originally requested and finally the wing structural analysis can take place.

9. Conclusions

In order to meet the technical challenges set for the next future of air transport, the aeronautic community should put more focus on the development of new design strategies and advanced tools aiming at a more efficient exploitation of designer knowledge. Better aircraft design depends on the availability of more product knowledge since earlier phases of the design process, which is a critical condition that available tools and the current design approach struggle to succeed. Designers need more flexible and powerful tools to virtually access ideas, create solid conditions to implement multidisciplinary design, free time for creativity via automation of all the repetitive activities that enormously hamper the design process. KBE can provide the tools to capture and reuse product and process multidisciplinary knowledge in an integrated way, in order to reduce time and cost for engineering applications via the automation of repetitive design tasks and a systematic application of design best practices.

The interdisciplinary DAR group of the Aerospace Faculty of the Technical University of Delft has started since few years to commit to a new program of research and development activities aiming at the development of new integrated but flexible design tools able to capture and automate the design and analysis process of complex products form the aeronautic and automotive world.

The development of a new set of so called high level primitives represents part of the output of the last years of research in the object oriented analysis and modeling of complex products. These high level primitives provide a way to support designers shaping and assessing concepts they have in mind in a faster and more reliable way. KBE applications have been created and grown in a modular approach, which implement and exploit the concept of theses primitives.

The experience gained by DAR during the involvement in complex multidisciplinary design and analysis programs has led to the formalization and development of the Design and Engineering Engine. DEEs represent a possible answer to the need of industrial community of open and modular design and analysis systems able to take advantage of experts and design/analysis tools, which are not always to be found within the factory walls, but are often distributed on a world wide scale. The flexible and not monolithic nature of the DEE structure is
the key to guarantee a prompt integration of new and different analysis capabilities and methodologies, in order to adapt to the different nature of design cases, and give the possibility to maintain the system and upgrade it during time.

In this paper several references have been provided to a number of works of PhD and MSc students of DAR, to give readers the idea of the large involvement in this research topic and the variety of study cases where the use of KBE techniques and the development of a DEE have been successful.

We acknowledge that financial support to the research topics discussed in this paper has been provided partially by the European Commission, under the GROWTH Programme, for the research project “MOB - A Computational Design Engine Incorporating Multi-Disciplinary Design and Optimisation for Blended Wing Body Configuration” (Contract Number G4RD-CT1999-0172), and by the Dutch Technology Foundation STW for the research project “Parametric Modelling and Meshless Discretisation Methods for Knowledge Based Engineering Applications” (Contract Number DLR.6054).