SCENARIO BASED AIRCRAFT DESIGN USING KNOWLEDGE BASED SOFTWARE METHODS

G. Dirks*, A. Schneegans**
*DaimlerChrysler Aerospace Airbus GmbH
**PACE Aerospace Engineering and Information Technology GmbH

Abstract

In a first part, a general introduction to the 'Technologie-Navigator' project is given. The complexity of decisions to be made in the context of technology application is illustrated using an engine integration example. Obviously, an integrated, coherent model of the complete decision chain is needed in order to come to thoroughly balanced, robust decisions about technology application. A brief description of the general approach of the methodology and the various part-models interacting with each other under conditions set by a scenario technique leads to the identification of requirements for the software-kernel. A description of the software engineering platform 'Pacelab' matching these requirements forms the second part of this paper. It is shown, why a new software platform is crucial for the realisation of the project. A description of the 'Pacelab' program system is given along with new requirements arising from the 'Technologie Navigator' project. The key software technologies which shall serve as the building blocks of the new software engineering workbench are presented and their key features are discussed. Finally, an overview of the current status of work is given along with a brief description of the existing software modules, the existing technology demonstrators and their future integration as well as their contribution to the future aircraft design system.

1 The 'Technologie Navigator' Project

The '*Technologie Navigator*' project was started 1999 at the Future Project Office of DaimlerChrysler Aerospace Airbus in Hamburg. The project is supported by the German Federal Ministry of Economics and Technology (BMWi) and is scheduled to be completed in 2002. Several partners from science and industry are contributing to the progress of the project. Common goal of all activities conducted in the frame of the project is an enhanced process of the integration of novel technologies in future aircraft projects [1].

The incorporation of a catalogue of probably available technologies and prognoses about their impacts on the design of future aircraft projects is one portion of the work planned. Coupling mechanisms between these prognoses and existing preliminary design tools for various disciplines (like weights, aerodynamics, performance, cost, economics, operation…) will be identified and appropriate models will be built and added to the existing codes.

A second part of the work is formed by the development of the methodological background for an enhanced design process. The tools performing a synthesis of the outcomes of various disciplinary tools must be adapted to cope with the integration of uncertain technological impact. But, talking about future aircraft projects, not only the actual progress of technology is hard to define. The assumptions about market development, the competitor's moves, future economic conditions and many others have proved to be at least uncertain, too.

The exemplary application of the newly developed methodology and the incorporated tools in an airplane project builds the final portion of work in the '*Technologie Navigator*' project and is scheduled for late 2001.

Today's project status is characterized by the recently completed definition of the methodology to be applied and the specification of the requirement for a software-platform linking the elements of the methodology. Several stand-alone demonstrators for different tasks have been developed or modified from existing tools.

2 The Need for an Enhanced Technology Integration Methodology

In order to cope with an increasingly competitive transport aircraft market, development decisions with critical impact on the long-term strategy of the aircraft manufacturer have to be carefully assessed. As new aircraft are usually designed within a product family context, which nevertheless shall allow for short-term reaction on new competitor aircraft or changing customer requirements, the introduction of technological improvements into these designs has to be coordinated with the development schedule and be evaluated for several scenarios about the things to come.

For instance, the expected further aggravated regulations of the environmental impact (noise, emission, climate, consumption of resources) of air traffic must be matched by the infusion of appropriate technological enhancements.

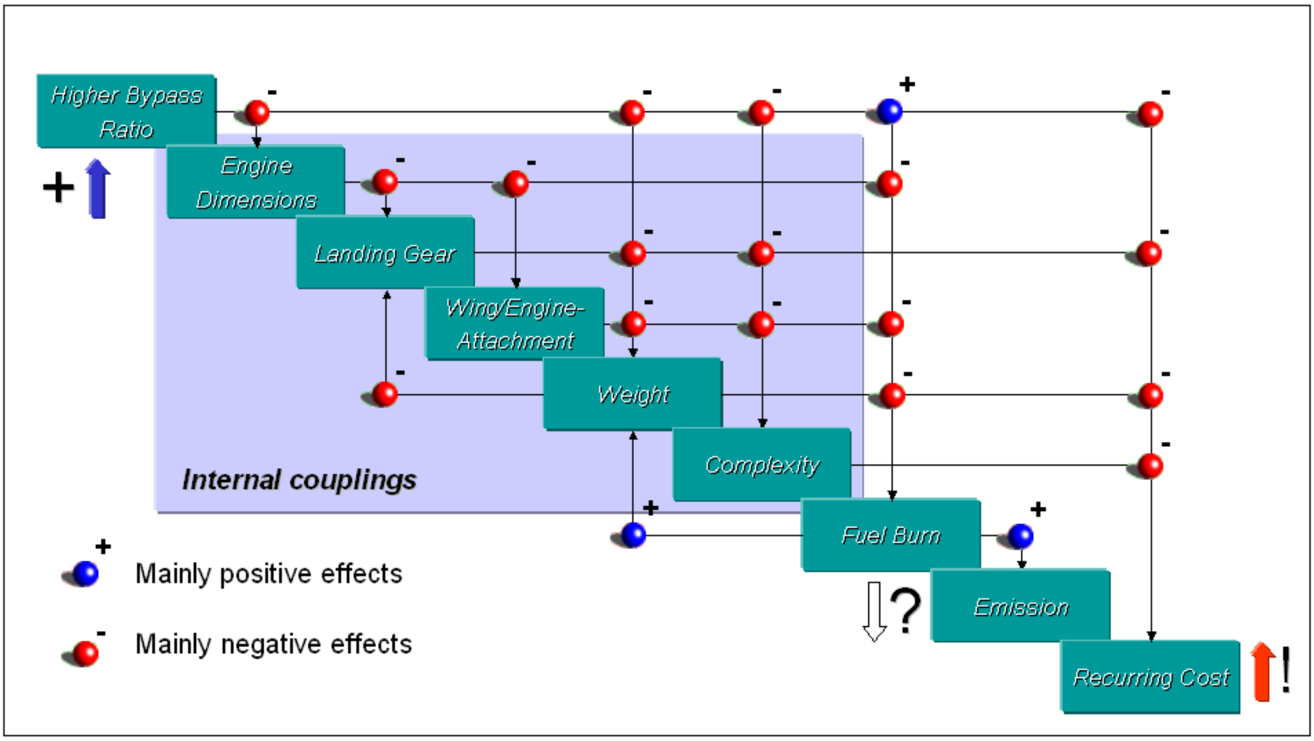

Fig. 1: N²-Visualisation of the engine integration example

One major step towards an environment-friendly aircraft can, of course, be expected from the introduction of advanced engine concepts, providing less noise as well as lower emission, thus diminishing the influence on the climate. More efficient engines would even reduce the consumption of resources and enhance the economic performance of the aircraft via decreasing fuel burn.

Unfortunately, along with the benefits, some drawbacks have to be considered. At first, the most probable way to enhance engines in the described way, will be a further increase of the bypass ratio, resulting in more complexity (geared fan ?), weight and bigger dimensions. The increased diameter will lead to a worse aerodynamic and structural attachment of the engines to the wing and probably require a higher landing gear to maintain ground clearance. Weight, complexity and dimensional increase of those (and many other) components will diminish the effectiveness of the solution not only at the propulsion unit itself, but also on the complete aircraft design. Whether the positive environmental effect would pay off in the end, has to be proven yet. In Fig. 1, a coarse N²-visualisation of these interactions is given.

Up to now, the potential benefit was considered only in terms of the aircraft design itself, not on the entire fleet flying. Any assumable scenario for a future emission policy (world wide or on a regional basis) will not focus on the emission characteristics of the aircraft design, but on the cumulative emission of the entire fleet of an airline, a nation, an airport or even a manufacturer. In this context, not only the ecological characteristics of the airplane design, but also the number of aircraft flying is of interest. The environmentally best aircraft design does not change anything, if it can not be sold. Thus, the economical performance as well for the manufacturer as for the operator cannot be ignored while enforcing the ecological aspects.

A second look on the consequences of this technology introduction effort brings up further questions. Talking about aircraft families and commonality aspects, we have to ask for the possibilities of adapting the new engine concepts to other thrust levels required for other members of the family, for maintenance commonality of the core engines and much more. Will there be a retro-fit option for older aircraft?

All these (or similar) questions are considered while deciding about the application of one proposed technology in one aircraft project. Deciding about the application of several technologies simultaneously, the 'normal' engineering judgment comes to its limits. Thorough decisions about the development of the technology portfolio itself, usually incorporating very long lead times, cannot be made at all without a sound knowledge about the characteristics of the couplings between all effects described before - and many more. For this purpose, an integrated, coherent model of the decision chain covering all these aspects is needed.

As not all couplings can be determined in advance, scenarios about the uncertain future come in place. The methodological linkage of different portions of the decision chain and the scenario technique will be developed in the frame of the project. A first step towards this goal was the definition of various models to be integrated in the enhanced technology integration process and the resulting requirements for a software platform. These will be described briefly in the next paragraphs, while the rest of the paper focuses on the software concepts identified to be suitable in setting up the mentioned linkage.

3 Elements of the Methodology

The general approach of the methodology is the interaction of dedicated models in various processes. These models are fed with knowledge from the 'normal' specialist's toolset. After compiling all information available to an overall model of the space in which the decision has to be made, several integrated process handling modules allow the exploration of consequences of different inputs or assumptions. The most important elements of the methodology are illustrated in Fig.2.

3.1 The models

3.1.1 Aircraft model

A model of the characteristics of a single aircraft is fed by the innermost design loop of the new methodology. The information stored in this model is the result of the application of 'normal' aircraft design tools. Classical computer based design methodology focuses only on this part of the overall process.

3.1.2 Family model

Nearly all civil aircraft models under production today are embedded into a family concept. Several advantages for both sides, operator (reduced training expenses for flight crew and maintenance, attractive contractual options etc.) and manufacturer (decrease of development and production cost, shorter time to market for new derivatives) have led to this situation. The family of aircraft is described in a model including the planned members (e.g. in terms of their placement in the range/capacity-space, design weights) and their sequence of Entry Into Service dates (EIS), the geometric and structural provisions needed for the first members, penalties resulting from following members or predecessors in the family, recurring, non-recurring-cost and operating benefits etc.

3.1.3 Technology model

For each candidate technology, a model of its impact on the aircraft, its development, production or operation is fed with the values proposed by the promoting technologists. Provisions for the consideration of uncertainty and incompatibility with other candidates are made. A special focus of this model lies on its variation with time.

3.1.4 Risk assessment model

The assessment of risk will be managed by a model that interacts with all values assumed to be uncertain in some way. The propagation of these partial uncertainties to global risks is done

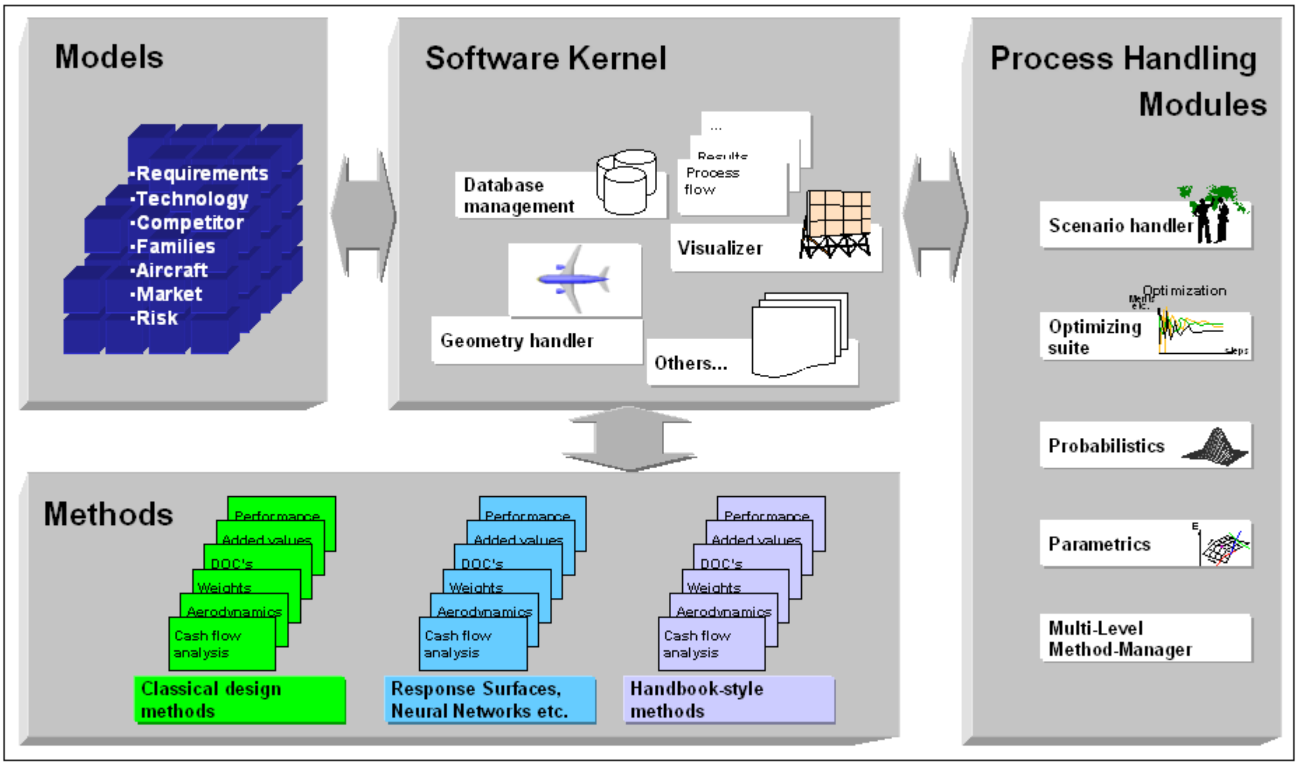

Fig. 2: Elements of the methodology

by the probability module (see below). The assessment of risk and the appropriate reaction to cope with that risk is managed by the risk assessment model.

3.1.5 Other models

In addition to the models described above, several other models, acting in similar roles are defined. Examples for these predefined, but not yet developed models are a market model, one for the competitor(!) and one dealing with evolution of requirements.

3.2 The methods

For the various disciplines involved in the early stages of aircraft design, computer based methods for the prediction of properties are available. The accuracy of theses predictions usually is a function of the complexity of the methods applied, the effort spent and experience previously gained. In turn, decreased demand for exactness of results usually comes along with decreased expenses in terms of data preparation as well as calculation time.

Depending on the problem to be solved in the frame of this project, different levels of accuracy for calculated results are acceptable. Therefore, different levels of fidelity of the tools can be identified. The 'Multi-Level-Method-Manager' described briefly below helps managing these various levels.

3.2.1 The 'Legacy'-Codes

The existing industry methods form the 'high end' of the accuracy hierarchy. The experience of the programming specialists was coded into these systems. The fidelity of the methods is 'best practise' and the knowledge about applicability and constraints of the methods is usually very good. As this kind of code has been developed over the years, the architecture and user interfacing is often 'old fashioned'. However, this type of code is the 'backbone' of the design process and should therefore be integrated in the methodology in the best way possible.

For rapid exploration of the design space as proposed by scenarios about the future, they may not be well suited.

3.2.2 Response Surface Methods (RSMs)

RSMs form the second level of accuracy. For RSMs, a purely mathematical approach is used to interpolate known characteristics of the design space. Simple forms of RSMs utilize e.g.

quadratic multi-parameter-polynomials. No knowledge about the physics or economics of the design problem is used to set up the form of the RSMs. The coefficients of the RSM-formulas for instance can be derived from dedicated calculation runs ('experiments') of the legacy-codes. The RSMs are very robust, having a scalable accuracy depending on the effort spent on the coefficient-generating experiments. No reliable predictions about the accuracy outside of the experimented region can be made.

Another example for this type mathematically driven robust exploration methods are 'Neural Networks', trained by the experience of previous calculations.

3.2.3 Handbook Methods

Classical hand book methods are a combination of a rigorously simplified physical model and a heuristics-based set of coefficients. The heuristic portion of the methods can be adjusted by experiments, similar to the approach applicable on RSMs.

Handbook methods can easily be adjusted to a reference point and provide good robustness with limited, but predictable accuracy due to the physics- based approach.

3.3 The Process Handling Modules

The interaction of the models, each querying the methods for the generation of information will be steered flexibly via process handling modules. Examples for these modules are :

3.3.1 Optimising suite

Using the object oriented software architecture described below, the implementation of optimisation algorithms or parametric survey procedures can be realized in a flexible way. A still ambitious task will be the development of an appropriate user interface for the definition and monitoring of the optimisation problem itself.

3.3.2 Multi Level Methods Manager

Especially the RSMs and 'Neural Networks' depend on information provided by 'higher' methods. Usually this will be done through the automated triggering of dedicated calculations, known as 'Design Of Experiments'. The execution and monitoring of these processes will be controlled by the Multi Level Methods Manager. Furthermore, the calibration of all methods to match given results and to provide smooth transitions between the methods of different accuracy will be supported by this module.

3.3.3 Probability Module

As mentioned before, the aggregation of local uncertainties to global risks for the design task will be handled by a dedicated module managing the aspects of probability. The methods under consideration lean against the methodology published by D. Mavris [2].

3.3.4 Scenario Handler

Scenarios help modelling a 'syndrome' of uncertain inputs. The development of the scenario handler itself has not made too much progress up to now, but as the scenario handler will serve as a 'feeder' for the parameters of the models, the known structure of the models was used to formulate the corresponding requirements for the software-kernel as well as for the scenario handler. This concurrent approach was necessary, as both, the development of the scenario handler and the completion of the software-kernel have considerably long lead times.

4 Required Software Architecture

Given the number of existing aircraft design systems in research organisations and industry, and regarding the analysis codes currently used in aerospace companies, one has to seriously justify the investment in the development of a new generation design system.

From an observer's point of view, there is no functional difference between the here presented Knowledge Based Engineering (KBE) software components, or existing legacy code and commercial systems. Each of those tools contribute with known precision to the finding of engineering solutions, hence, integration of existing legacy systems through dedicated bridging software is popular among engineers. However, providing integration interfaces is by far not enough. This strategy will only lead to better management and exploitation of existing technology but will neither provide for new ways of engineering software development nor for more stable and robust solutions. The software engineer who has to manage new engineering software development will still be left alone with defining the architecture for new components. The challenge of a new software architecture is hence to integrate existing analysis code and to provide a platform for development of new and robust, knowledge based engineering objects.

4.1 Pacelab Engineering Workbench

The *Pacelab* system has evolved from the R&D project '*Flying Objects*' which aimed at providing software components for aircraft modelling and analysis based on new software technology. During its development the system has undergone a significant shift into a more abstract architecture, which does not specifically address aerospace problems but intends to provide for pre-built components and standardised interfaces suitable for supporting development of engineering software in general.

The system has successfully been applied to the development of aerospace software solutions in the fields of aircraft performance assessment and cabin interior layout and definition. Moreover, it has been used in major in-house development projects of leading aerospace companies. The '*Technologie Navigator*', however, requires some significant enhancements to the current system. Besides the development of dedicated components for the involved design disciplines and the process handling modules, this mainly refers to the development of new data abstraction models and the required interfaces to couple the system to existing or to newly developed analysis code.

This is especially important regarding the complex demands of a scenario handling module which is still in its definition phase. Hence, the underlying workbench has to be designed with enough potential for future changes and enhancements.

4.2 Geometry Model

The geometry model of the aircraft design system as the main data supplying element in the design synthesis chain needs to cope with some key requirements:

- High degree of detailing
- Extension flexibility
- High computation performance
- CAD compatibility

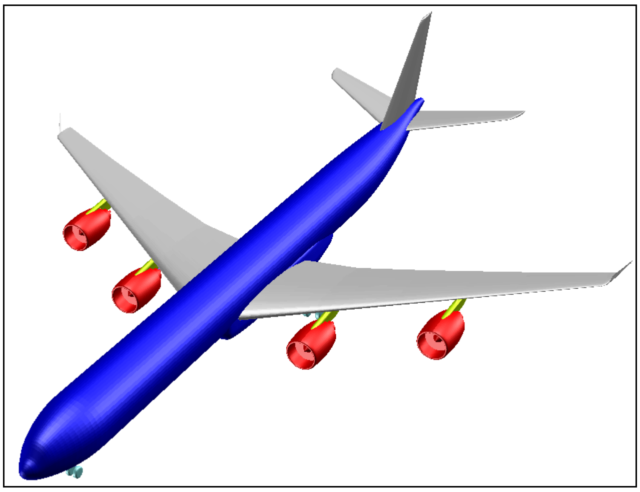

Fig. 3: Geometry model

In order to satisfy the latter two requirements, the model will be built as the combination of a numeric, parametric model based on profile variant programming and a high-end, 3D-CAD model. The numeric model will be inquired by the system to the extent possible, exploiting its computational speed. The CAD model will serve for complex analysis like wetted or projected surface calculation as well as volume calculation. Furthermore, it will empower the system to model geometric bodies as the result of Boolean operations of other bodies, which is the case for belly fairings or similar other aircraft components. The CAD module also enables the system to export its geometry to commercial CAD systems for further processing or alternatively to import specific existing geometries via standardised data exchange filters. Both the numeric and the CAD model will be integrated into the system via its object-attribute interface as described in [3].

4.3 Parametric Attributes

Pacelab's standard interface for integration of component attributes, see [3], requires the availability of the attribute software component at compile time. Attributes like the described CAD component have to be integrated this way, as they are too complex to be mapped into a simple form. However, most of the attributes one can think of in the scope of an aircraft design system have very simple data types and could thus be created at run-time. This is the case for single-number values like a weight attribute, but there are also ways to create records of data entries to a more complex, multi-value attribute at program run-time.

Such a user- definable, parametric attribute will be a powerful yet simple building block, capable of incorporating the bulk of flexibility required within the '*Technolgie Navigator*'. *Pacelab* will provide for a user interface for the definition of these attributes. It will give means to add simple data types to the value-container and will require the user to assign the attribute a unique name. The attribute will then be stored in a repository for subsequent use. A powerful feature will be the integration of a script interpreter which not only allows for rapid coding of analysis procedures, see 4.4, but which will also allow for creation of simple user interfaces.

4.4 Analysis Code Integration

Evaluation of the user-definable attributes will be performed through a transparent interface, which will give the following options:

- Setting of fixed values
- Assignment of an executable script
- External program execution through operating system calls
- Coupling of legacy code interfaced through *DCOM* or *CORBA*[1]

The interface is an integral part of the parametric attributes as described in 4.3, and its flexibility is the technical basis for the implementation of a multi-level method handler as required by the '*Technologie Navigator*'. Responsibility for suitable combination of the methods to a consistent and meaningful method set will be left with the aircraft design specialist.

4.5 Knowledge and Process Modeller

The '*Technologie Navigator*' requires two, apparently contradictory design goals for the underlying software system. On the one hand, the aircraft design loop shall be a robust, inner-cycle process which automatically responds to changes in the parameter space of the scenario handler. On the other hand, it shall be flexible enough to be configured for different tasks and analysis modes.

On the process control level, for example, it may be necessary to switch the synthesis components on or off, like automatic scaling of the engine, wing positioning or tail sizing. On a more detailed level, the engineer might want to change certification regulations by selecting a new regulation 'package'.

Traditionally, this behaviour is realised by hard-coding a well defined scope of possible solutions. This approach is not sufficient for fulfilling the required flexibility. Since design strategies or economic and socio-ecological constraints are highly dynamic, the inflexibility of the 'hard-code' approach is an obstacle.

A solution to this problem is offered through what is nowadays understood as artificial intelligence. In contrast to the early vision of this information technology discipline, today's efforts do not focus on the goal of capturing human knowledge on a broad scale, but to concentrate on problems with a well defined scope. The central point is the concept of 'rules'. Know-how or, more precise, design knowledge, can usually be expressed in two parts, a situation (knowing when) in which it can be applied and an action (knowing what) that has to be taken.

This approach can be directly mapped to a design system's requirements, e.g. the control of processes and the reaction to infringements of constraints or regulations. The knowledge formulated as rules defines the rule base, which can be stored in a readable form separately from the application. The software system uses this rule base as an external source, so the software does not need to be changed with changes in the knowledge base.

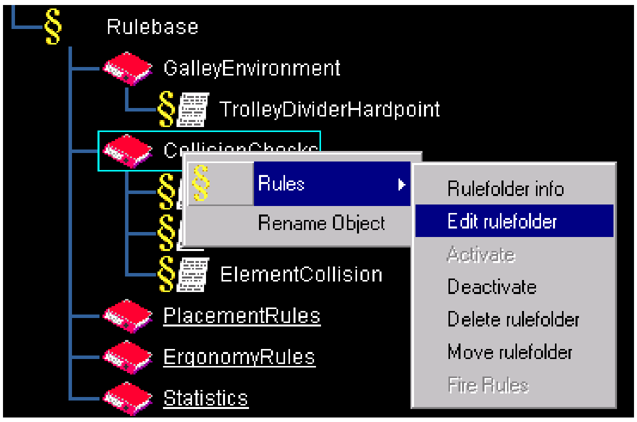

Fig. 4: Knowledge representation

A commercial rule-based system (ILOG Rules™) has been integrated into *Pacelab*. A dedicated control agent, which is aware of every status change during the design cycle, checks constraints and regulations.

[1] DCOM and CORBA definitions describe object communication interfaces for execution of network wide program resources.

Another agent automates engineers' design strategies formulated as rules. An intuitive graphical editor supports the formulation and modification of rules. Rules can be graphically combined to packages of arbitrary complexity which can be activated or de-activated at run-time through user interaction or through automatic setting, see **Fig. 4**. Existing software components can be set under 'knowledge control' with relatively little effort. The run-time performance which has been experienced in a customer-specific project gives a promising outlook to the application within a complex design system.

4.6 Data and Process Viewers

Pacelab offers data viewer for 2D and 3D geometry and diagrams. The Microsoft Internet Explorer is integrated as an ActiveX component and gives the system full access to the latest hypertext language functionality. Combined with an 'Output Generator', this hypertext facility allows the administrator or, if necessary, the user to define scope and format of the output, which can be a combination of text and graphics in print- and presentation-ready quality.

In the scope of the '*Technologie Navigator*', the diagram viewer will have to be extended for specific visualisation techniques which best satisfy the needs of a multivariate result parameter space.

It is intended to support the current approach of implementing knowledge through rules by an editor, which relies on a graphic representation of the processes rather than on list and input-field based methods. This viewer will hence not only sketch the design processes but give interactive access for their manipulation.

4.7 Status of Work

The current status of the software illustrates the component architecture of the *Pacelab* Engineering Workbench. Several basic components and technologies of the future aircraft design system are available and operational as stand-alone applications, which will be plugged together with the software modules which still have to be developed.

Although developed with the perspective of usage within a comprehensive design system, the components have not specifically been designed for this role but simply make use of the standardised interfaces of the *Pacelab* core system. These components and all future elements will find their anchor point in the system's tree structure, communicating via the data bus given with the attribute interface.

4.7.1 Geometry and Payload Definition

The current capabilities of the system regarding the generation of geometry can be presented within the cabin configuration system *Pacelab Cabin*. The program provides for a geometry model for fuselage-like bodies and offers comfortable definition interfaces which can be used without skill-intensive CAD knowledge. The software allows for modelling a wide range of aircraft, from business jets to multiple-deck commercial transports.

Given the outer contour of the fuselage, the cabin specialist can then define the payload compartments and insert cabin items out of an extensible database. *Pacelab Cabin* largely automates the positioning of the cabin interior items and consistently checks compliance with certification regulations and other rules of arrangement. Standard general arrangement drawings and reports are generated at the touch of a button.

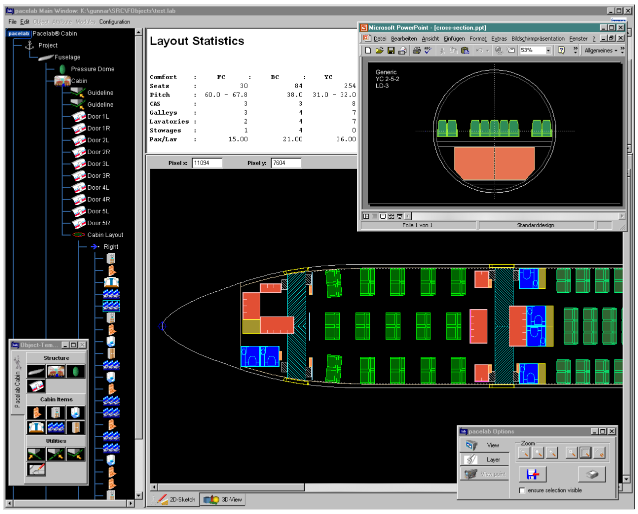

Fig. 5: Fuselage and payload definition

Based on this product, a knowledge based add-on product has been developed, which fully automates the development process of **P**assenger **S**ervice **U**nit (PSUs) design. The installation and wiring of all systems contained in the overhead stowage compartments and the associated supply channels is generated according to pre-defined and modifiable rules, the degree of automation is very high.

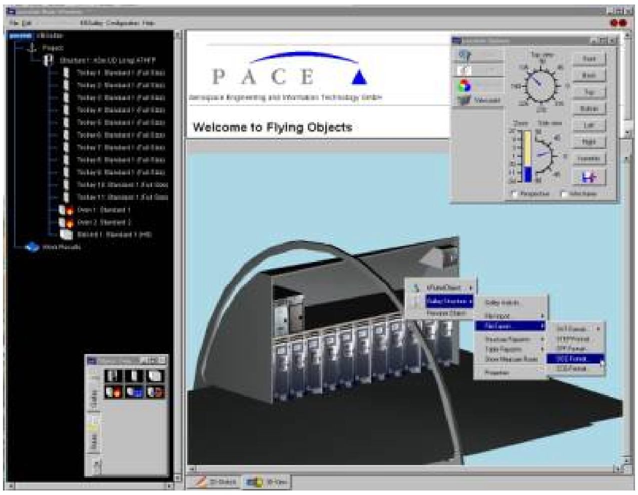

Fig. 6: Structural detailing and 3D modelling

As an example of how specific modules can increase the detailing of the design study process, Fig. 6 shows an application which focuses on the detailed design and analysis of aircraft galleys and which gathers the geometric entities of the galley structure from the multi-curved fuselage model. It is also a demonstration for the 3D visualisation capability.

4.7.2 Aircraft Performance

As an elementary part of the aircraft design and evaluation cycle, the fuel burn calculation for one or several design missions is a central component to any design system.

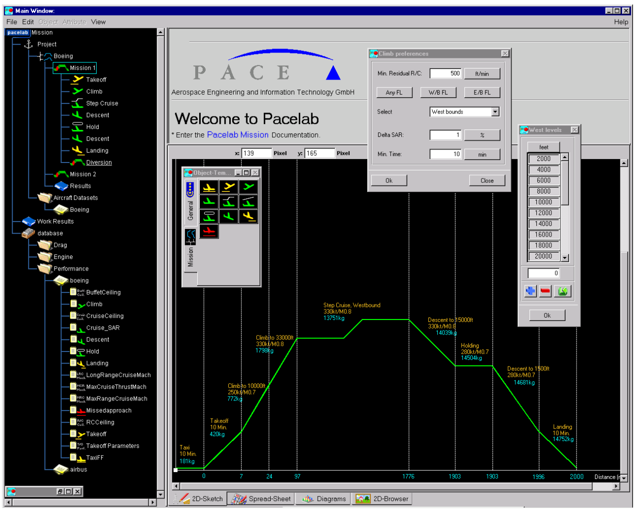

Fig. 7: Aircraft performance assessment

This analysis module is available through the program system *Pacelab Mission*. The system processes either integral performance data providing for aircraft manufacturer's data formats or by processing drag polars and engine deck data. The system is operational and suitable for simulation of environmental scenarios through its interfaces to geographically dependent atmospheric models.

4.7.3 Knowledge Engine & Browser

Pacelab's knowledge module is available in a first release. It is completely independent of the application built upon the engineering workbench and offers a well-defined, standard interface for interacting with the application-specific engineering objects.

This has been successfully demonstrated within a customer specific project, where a program for 3D galley design was enhanced by the knowledge processing module without major change of the existing program structure. The knowledge module allows for definition of galley specific design rules and for the formulation of constraints, which are automatically checked for violation. It acts as an agent, which monitors the design process and parameter space. Whenever conditions apply to the current situation or whenever a constraint is violated, the agent will execute the pre-set action.

5 Summary

A description of the project '*Technologie Navigator*' is given with focus on the methodological needs. The demand for enhanced flexibility of the software kernel to be developed was emphasised. Though not yet specified to the details, the need to incorporate a great variety of possible scenarios of the future was transformed into a specification for the software platform to be used in further development. The software engineering workbench *Pacelab* matches most of these requirements and provides the architecture needed to develop the missing components and features. Examples derived from existing applications using *Pacelab* are presented to illustrate this. The main software building blocks of *Pacelab* are both flexible and simple enough to cope with the upcoming demands of the scenario handler, which is concurrently being developed.

6 References

[1] Dirks G.A. *Selection of Technologies for Future Aircraft Family Concepts*, 1999 World Aviation Conference, San Francisco, CA/USA, AIAA 1999-01-5632

[2] Kirby M.R and Mavris D.N. *Forecasting the Impact of Technology Infusion on Subsonic Aircraft Affordability*, 1998 World Aviation Congress, Anaheim, CA, SAE-985576

[3] Schneegans A, Haberland Ch and Kokorniak M. *Flying Objects – An Object Oriented Toolbox For Multidisciplinary Design and Evaluation of Aircraft*, 1998 ICAS Congress, Melbourne, Australia, ICAS-98-6.4.4, 98-31680