AN OBJECT ORIENTED APPROACH TO CONCEPTUAL AIRCRAFT DESIGN THROUGH COMPONENT-WISE MODELLING

A. Schneegans, Ch. Haberland, M. Kokorniak, B. Domke
Institut für Luft- und Raumfahrt, Technische Universität Berlin, Germany

Abstract

Following the concept of 'right first time design', increasing emphasis is being given to the numerical design of aircraft. Apart from the conceptual design phase, modern design tools should cover the analysis of competitor aircraft and allow comprehensive assessment of technological and operational scenarios. Traditionally, research activities in aircraft design systems have focused on the development of precise analysis methods for all relevant aircraft design disciplines rather than on the introduction of modern software technology. Therefore a new design system architecture with workbench facilities is proposed which shall allow dynamic coupling of arbitrary analysis methods. A highly detailed but generic description of aircraft components is regarded as a basic requirement for such a system. Objects instantiated from such a superclass could provide for the flexibility to generate problem and design discipline specific decomposition schemes at run time. This is regarded as a key for the flexible definition of unconventional aircraft configurations. In order to evaluate the required procedures and software techniques, the design system *VisualCAPDA* is chosen as a testbed during development of the foundation class-library. Its *FORTRAN/C/C++* architecture offers a good testing environment and the possibility to aim for maximum reusability of existing *FORTRAN* analysis routines.

1. Introduction

Numerical conceptual design of aircraft is more and more regarded as a key opportunity for 'right first time' design. The flexible variation of important aircraft parameters in an early design phase and the assessment of their impact on aircraft performance and costs allow extensive configuration investigations before freezing the layout for a following design phase. Besides the definition of a new project, a design tool should also be applicable for the examination of already existing competitor aircraft. It should as well be able to validate developmental risks within different technical and operational scenarios. Hence, the design system should provide for a comprehensive range of query interfaces. In order to cope with emerging technologies, these interfaces should be conveniently adaptive. Advances in aerodynamics and propulsion, ecological restrictions and prospects of alternative or even exotic technologies have to be considered.

Ideally, a future design tool should support the investigation of every conceivable configuration, including multi-surface aircraft, flying wings, joined wings and lifting bodies. Depending on the accessibility of precise data, a design tool would have to offer different levels of analysis depth. For example, the system should handle integrated drag polars from external sources as well as measured performance data merged with internally generated drag curves. Otherwise the system should inquire its implemented methods ranging from top-level panel methods to bottom-level heuristic estimates. This flexibility requires a smooth and consistent integration of highly different analysis levels while simultaneously supporting the user through an advanced graphical user interface *(GUI)*.

Closely related to the support of differing analysis levels is the requirement for user-friendly extension of the system's method data base. The flexibility of the system should not be restrained to the integration of given numerical data but should also allow dynamic linkage of user-specific analysis methods. This implies a redefinition of the design system role. Instead of a highly specialized analysis tool aimed at specific problems, the system should be seen as a workbench, where arbitrary methods can be linked together. Like a docking module, the design system would provide the calculation methods with appropriate data, keep track of the design process through respective data storage and display configuration and performance characteristics to the user.

The support of different analysis levels and the requirement to provide user-specific analysis methods with input data implies the need for a thoroughly detailed description of the airframe, its components and properties. The computer internal modelling of parameters must therefore be a superset of the data required by all potential analysis levels.

According to the required scenario for a future aircraft design system, two basic fields of development have to be adressed in order to allow the early estimation and simulation of aircraft characteristics:

- Analysis methods, which give precise information in all relevant design disciplines, e.g. masses & structures, aerodynamics, performance and costs.
- Design tools, which flexibly allow the integration and interaction of analysis methods, manage input data, supply a complex but consistent computer-internal representation of aircraft properties, allow project management, control the calculation loops and visualize output.

Traditionally, aerospace research activities have strongly focused on the development of highly precise and robust calculation methods for all design disciplines. The range of structural analysis methods comprises semi-analytical / semi-statistical approaches as well as finite element methods or combinations of both. Aerodynamic methods vary from simple correlations to sophisticated panel-codes or Euler- and Navier-Stokes-equation solvers. Modern numerical analysis tools effectively exploit the computational power offered by state of the art computers in combination with modern operating systems and the quality of their output is steadily being improved.

Design tools on the other hand, which have to cope with complex data management and varying input schemes generally do not reflect the advances which have been achieved in modern software technology. Conventional design systems such as *CAPDA*[1] lack modern interfacing mechanisms like graphical user interfaces or extensive user-input validation. Their batch-oriented behaviour restricts the flexibility of usage while their usually procedural architecture is complex and can be difficult to maintain or extend. A recent release update called *VisualCAPDA*[2] already offers a much better handling quality through a graphical user interface as well as a comfortable analysis generator and project management. However, if problems like unconventional configurations - going further than just a three surface aircraft - should be adressed, or modifications other than the implementation of new calculation methods be performed, the limits of conventional program architectures become evident. These are not only due to their usually procedural architecture combined with corresponding coding languages like *FORTRAN*. A main obstacle for the flexible adaption to new problems and questionings is imposed by the rigid decomposition scheme of the synthesis, the optimization and the analysis process as well as of the aircraft itself.

Regarding modern approaches, e.g. concepts based on expert system shells[3], it becomes obvious that even though they do not impose a rigid structure for the solving process of a specific problem, their scope of application is nevertheless limited to what they were precisely designed for. The reason becomes quite clear when the aircraft design process is regarded.

The analysis of the design process within a design team composed from different departments shows, that every design discipline requires their specific decomposition scheme[4]. Moreover, one can show that decomposition schemes already vary from one technical concept to the other (conventional aircraft → flying wing). If differing analysis methods are to be applied, the problem of choosing a decomposition scheme becomes even more complex.

Design systems which rely on a single decomposition scheme thus have to be a compromise which consequently shows its limits whenever unforeseen configurations or problems are investigated. A dynamical adaption of the decomposition scheme with respect to aircraft configuration, analysis methods and design disciplines could thus show a way towards a greater flexibility and an improved adaptability of a future design system.

A new approach is proposed, which is based on object oriented component modelling on a highly abstract description level. Generic description of aircraft components disregarding the specific function of every single component would give the possibility to put an aircraft together out of a toolbox. Each design discipline would have to request the components for their contribution to its result. Additional 'container classes' would have to be introduced for combination of components into assemblies or functional groups.

Several techniques and procedures have to be introduced for the development of such a system. Their testing would be difficult or even impossible without relying on a testbed, which could provide for basic functionality. The *VisualCAPDA* design system offers a good flexibility for extension and is therefore chosen as a technology demonstrator. Additionally, its hybrid *C/C++* and *FORTRAN* architecture allows to integrate a new object oriented approach whilst re-using existing *FORTRAN* analysis methods at a great extent.

2. Objectives of a Future Design System

Overall aim for the introduction of a new aircraft design system architecture rather than the improvement of existing systems must be a substantial increase in flexibility and computing capabilites. In [2] it has been shown, that the evolutionary introduction of modern software standards can significantly improve the scope of application of an existing and well-tested conventional system. Nevertheless, the limits of a standard architecture become evident, e.g. when unconventional aircraft configurations are to be calculated. The adaption of a conventional system would require extensive modification of the program code in every single design discipline. The introduction of a powerful database, the coupling to a *CAD* system or the introduction of network capabilities would hence certainly overload a conventional approach.

With regard to the required technological improvement of a new design system, the following objectives are defined, giving a basis for the development of a future design system.

2.1 Configurational Flexibilty

One major goal of a new design system must be the capability to flexibly assist the definition of arbitrary aircraft configurations, see figure 1. For this purpose, a very flexible description of the aircraft geometry and its characteristics has to be introduced.

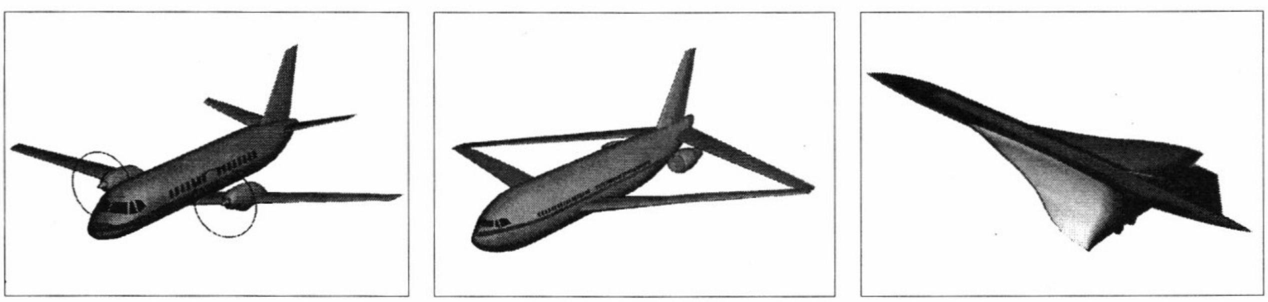

Figure 1: Configurational Flexibility

The physical and performance related properties have to be generalized in a way which allows to assign them to every single component. This description would be compliant with the defininition of conventional aircraft configurations as well as with unconventional ones.

In case of a flying wing, additional features which are generally assigned to a fuselage would be added, e.g. enabling the modelling of passenger compartments. Data structures and procedures could be identical at a great extent, a wing or a fuselage component could hence be derived from a common superclass. From the computational point of view, a flying wing would no longer be 'unconventional' but just a regular component with enhanced attribute behaviour.

2.2 Open System Architecture

The system architecture has to offer open interface capabilities. This feature should cover several fields, wherein the dynamical linking of calculation methods represents a decisive aspect, as this allow to extend the method database without manipulation of the core system. With regard to concurrent engineering strategies and department-wide workgroups, this capability would allow to integrate design discipline specific calculation methods without the need of recompiling in a probably heterogenous networking environment. Dynamically linked user specific methods should instantly be selectable through a graphical user interface in order to ensure a high level of user guidance.

Import filters should be provided to offer numerical data integration capabilities. This should be applied for the import of aerodynamic data (calculated or from test flights) as well as engine card decks. Once imported, these data should be stored in the internal, object oriented data base. The dynamical linking of user specific methods and the import of numerical data into perhaps centralized object oriented databases points out the need for a client server based networking architecture. ***Remote Procedure Calls*** (RPC) are regarded as the appropriate way of integration, relying on approved standards of UNIX or PC-based environments.

2.3 Flexible Query Capabilities

The query capabilities of a future system should significantly be enhanced. Systems like *Visual-CAPDA* [2] offer a limited selection of queries, which can of course be extended, albeit with substantial programming effort. For this reason, the program developer is generally lagging currently required program features. A browsing tool which lists parameters, either basic or compound, could, if coupled with a flexible table and chart generator, provide for the required flexibility. The program internal resource management which would be necessary for such a system could be provided by inferencing mechanisms as used in *Constraint Logic Programming (CLP)*. These techniques could also favourably be applied for configuration management, e.g. in seat distribution algorithms.

2.4 Integration into CAD Environment

The integration of either the design system or the produced output into a *CAD* environment is of significant importance. Without a smooth and consistent integration into an existing *CAD* system, the uninhibited dataflow of the design process within a team cannot be ensured. The integration could be performed in different, alternative ways. Besides the conventional data exchange via parameters or standardized data exchange formats (*IGES*, *VDAS*, *STEP*), the design system could directly be integrated into specific *CAD* systems, provided they offer appropriate interfaces. As a result of the generalized, object oriented structure, the geometry interface could be coupled dynamically to the *CAD* system via *RPC* or pipe communication. Providing a flexible graphical user interface, the user could have the impression of working solely with the *CAD* system, using the design system just like an extension. Whether the design system should offer the entire functionality necessary for the graphical initialization or whether it should rely on the functionality of a given *CAD* system, would depend on the work flow of a specific design team and on the capabilities of the *CAD* system. In any case, the toolbox character of a future design system should easily give both possibilities for the integration into the design process.

3. Problem Decomposition in Aircraft Design

3.1 Decomposition Strategies

Component and design process decomposition is applied throughout the entire process of aircraft design. Decomposition schemes can be found for every single design discipline either on a component level or applied to design and analysis processes. Overall aim is the reduction of a problem to a discernible and computationally handable problem. For this purpose, skeleton models of aircraft components, assembly groups or even processes and algorithms are defined. The resulting decomposition schemes vary depending on the design discipline they are applied for and on the abstraction level of the calculation methods used therein. Moreover, the decomposition schemes can depend on the configuration of the aircraft itself.

Two basic approaches for the definition of a scheme can be identified. The physical-morphological approach intends to subdivide an aircraft into its elementary parts. It is mostly used within assembly mockups for concurrent *CAD* systems. Aircraft design programs rather choose a design-discipline/attribute-oriented approach. Due to the static data structure in conventional programs, a compromise between the considered design disciplines is defined which has to provide very different views on a specific aircraft configuration. A structural department, e.g. would subdivide an aircraft into primary (load-carrying) and secondary structure. The components would be classified according to their geometry and load distribution and treated as flat or curved plates, beams, etc.. A construction oriented department would rather group components to assemblies, which correspond to construction process units, while an aerodynamic department would only consider wetted components, regarding outer geometry and lift/drag characteristics.

A design-discipline approach implements the knowledge about the aircraft configuration within a discipline module rather than in the component. In *VisualCAPDA*, e.g. the aerodynamic module manages the data handling required for the calculation of aircraft aerodynamics for conventional, canard or three-surface aircraft. The corresponding knowledge is required for the graphical user interface which is responsible for geometric initialization. System enhancements which aim to introduce further configurations like joined wings, double-deck fuselages, multi-fuselage aircraft etc., hence imply the modification of large amounts of code in every design discipline.

The problem to define a fully satisfying decomposition scheme is illustrated with figure 2 and figure 3. Figure 2 is a decomposition scheme which could be used for a seat-layout and fuselage design algorithm. Figure 3 is a typical, simplified decomposition as used in many aircraft design systems. Both approaches would differ significantly if they had to describe a flying wing. Payload compartments then would have to be a part of the wing structure. The position of components missing in the simplified structure in figure 2 is disputable. An engine nacelle could be part of the latter, but also form a group together with fairings or similar components.

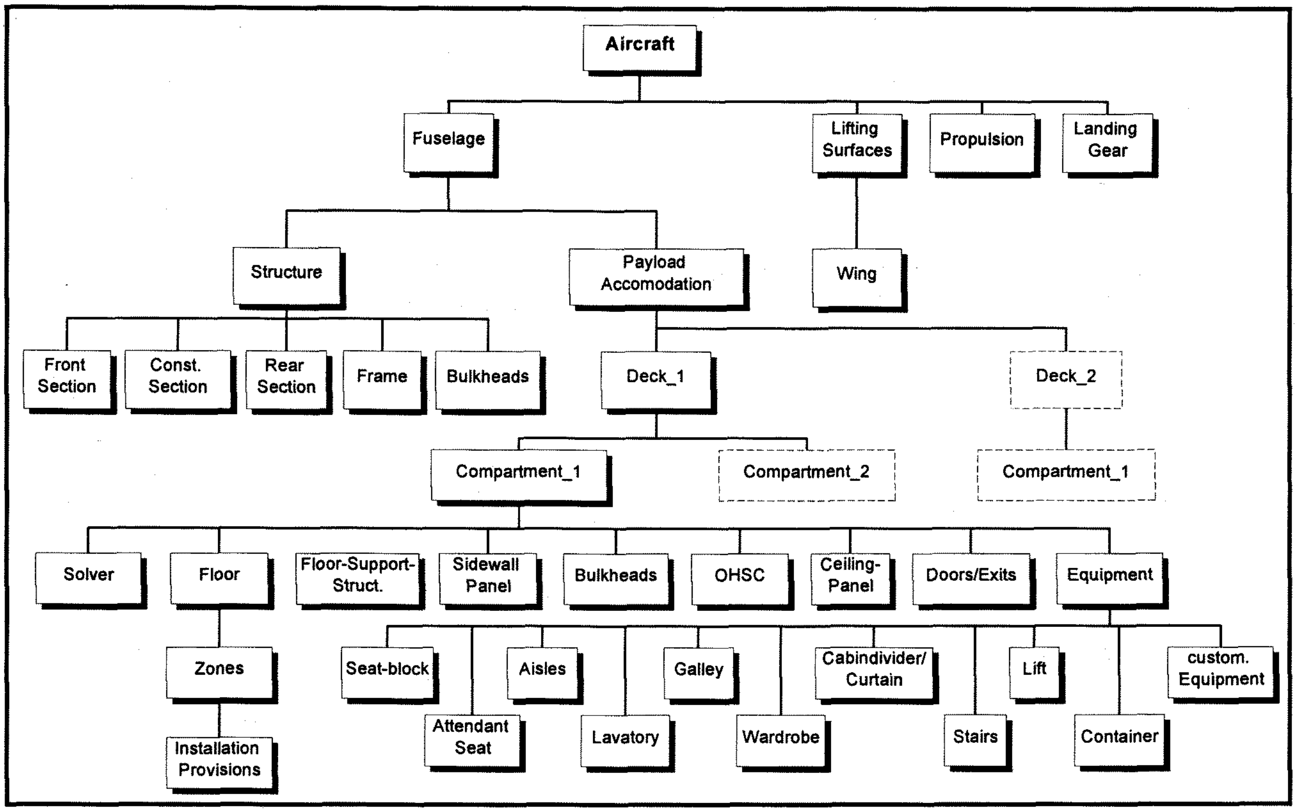

Figure 2: Sample Decomposition for a Cabin Layout Tool

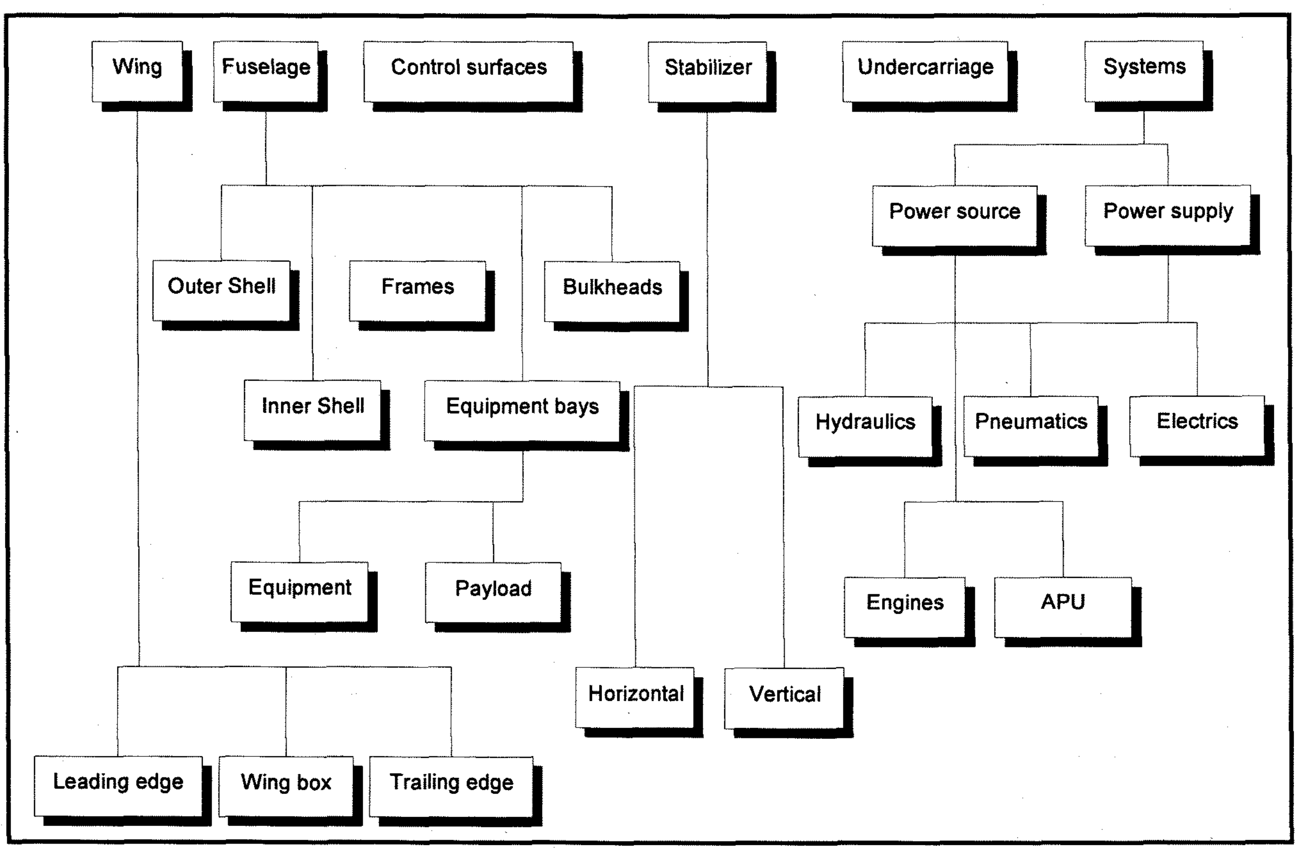

Figure 3: Simplified Decomposition for Aircraft Design System

3.2 Limits of Conventional Architectures

One major problem of decomposition schemes lies in their static nature. A decomposition model for aircraft components is valid for a specific aircraft configuration, under the view of a specific design discipline as well as under consideration of the applied method set and its abstraction level.

Design systems based on non-adaptive schemes show several disadvantages. In [5] it is shown, that the application of one single decomposition scheme to the design synthesis and optimization process does not entirely satisfy the needs of functionality, computational speed and exploitation of networking environments. Hence, a strategy for the dynamical definition of decomposition schemes for optimization processes is presented. Beyond any optimization purposes, the application of a single decomposition scheme restricts the flexibility of the entire design system. The definition of arbitrary aircraft configurations, the graphical manipulaiton of the aircraft geometry as well as the application of analysis methods at different levels of analysis depths is tightly linked to a rigidly integrated decomposition model. Any extensions or enhancements within the mentioned fields require intensive programming effort, simultaneously increasing program complexity and maintenance needs. In combination with generally applied procedural programming languages, this architecture is not regarded as the appropriate solution for the development of a future design system.

4. Object Oriented Decomposition

4.1 Component Modelling

Instead of imposing a compromising decomposition scheme, the new approach intends to establish an object oriented class-hierarchy for aircraft components. Generic component description could provide for a flexible means to assemble arbitrary aircraft configurations out of a prepared toolbox. By providing generalized component descriptions with virtually defined attributes and functions, instantiation at run time could assign or neglect specific attributes disregarding the functional role of the object. So defined aircraft components would be 'self-reporting' and thus transparent with respect to their physical behaviour and their data structure.

An aircraft component 'superclass' would have to be defined, which would provide for the interfaces required for every design discipline. For geometric component description, an interface relying on dimensionless parameters would be applied[6]. This technique has been succesfully introduced in Visual-CAPDA[2] and its ancestor. In order to allow a more powerfull description, this approach would be enhanced by a modelling based on a dynamically choosable number of cross-sections and the respective meridians used as guidelines for their longitudinal alignment. The cross-section model would be handled as an exchangeable method, thus allowing arbitrary geometries. In case of cylindrical bodies a multi-hyperelliptic description would give the flexibility to model fuselages, engine or propeller nacelles, tanks, external payload etc.. For the modelling of lift-producing components like wings, tails or similar functional units, this description could be exchanged for an x-y scatter modelling blended by *NURBS* or *BSPLINES*. Any other model could of course be integrated, too. The geometry methods would be based on an internal, commercial *3D*-geometry modeller. Components could be defined as solid bodies taking advantage of the computing capabilities of modern geometry modellers. Alternatively, the dynamic coupling of the geometry methods could be applied to the linkage of a *CAD* interface. This technique has been tested for the implementation of a *CAD*-based cabin layout module. As described in chapter 2, this would provide for the possibility to seamlessly integrate a design tool into an existing design environment.

Apart from the geometric description of the aircraft which gives substantial information for every other design discipline, the generic description would also provide their respective interfaces. Weights, lift&drag characteristics, thrust, temperatures, etc. could easily be assigned to each component. The design disciplines would then have to 'collect' the contributing attributes from each component. Container objects could handle this problem, also allowing to group components together to design discipline specific clusters. In case of a weight breakdown, e.g., a container-class would collect the weight items listed in a top-level item-list. Members of this list would be all those components which would contribute with their own weight estimation method or a fixed numerical value to the breakdown. The groupment would be dependent on the selected method set. A simple weight estimation approach might for instance supply the weight for the entire tail-group, the group of vertical and horizontal tail would hence be handled as a virtual aircraft component.

Not only physical and geometric properties, also configurational information could be provided by the components themselves. The attribute 'attached to' might describe the structural linkage between a wing and a fuselage by the respective forces and moments. This information could be used by high level calculation methods without inhibiting a simple method from getting its required information. Overall aim of the proposed approach is hence the supply of as much information as possible in order to satisfy the needs of the selected method set simultaneously giving the possibility to assign every attribute under consideration to any component.

multitasking facilities. Time consuming optimization processes would be processed parallely to the current working session without loosing control over the initiated processes. An effective project management would keep track of the project stage, offering revision information and ensuring the consistency between initialized and calculated configurations. To-

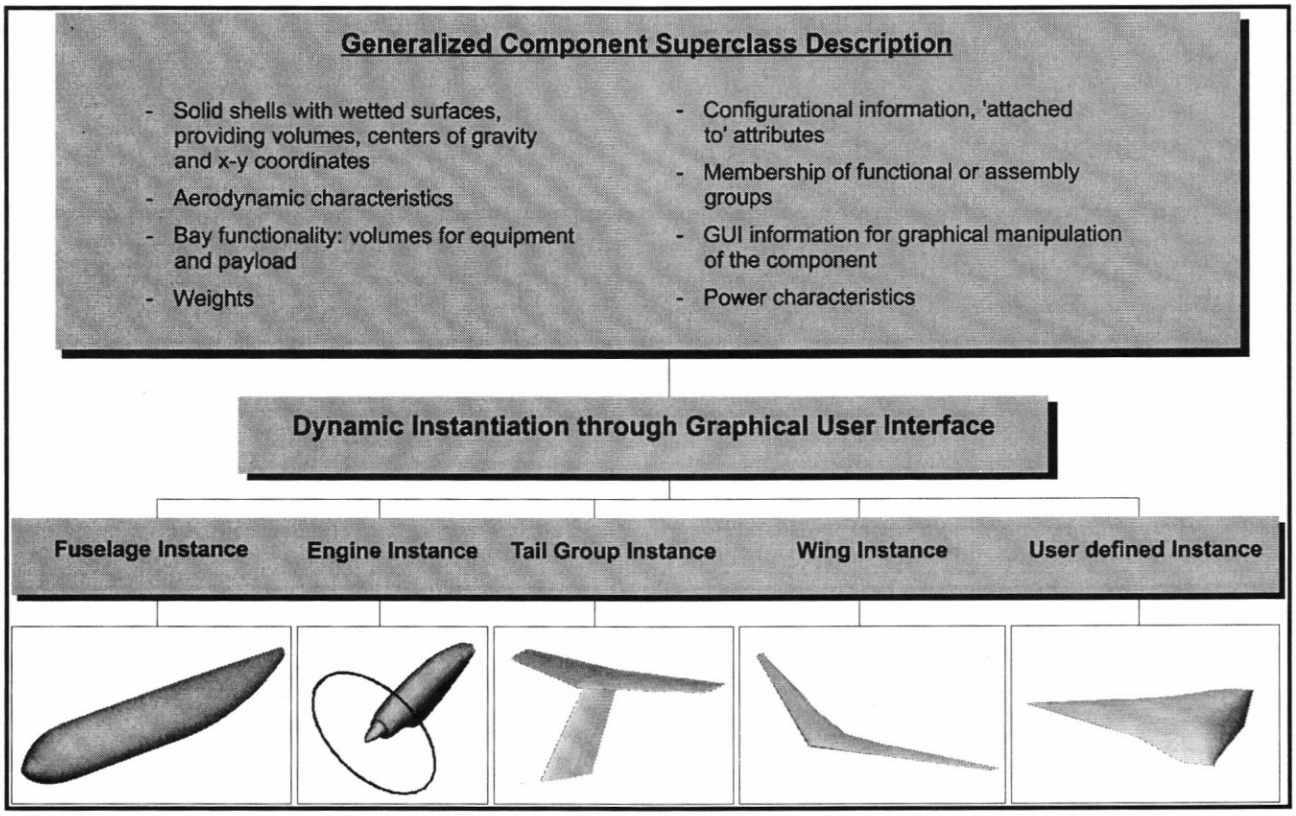

Figure 4: Class Properties and Derived Objects

Figure 4 illustrates the modelling capabilities of a first, prototype approach for aircraft components. Of course, graphical user guidance represents a crucial point for the proposed system. Extensive graphical support must be supplied, routine initialization tasks be supported by an object oriented component data base. This would cover the flexibility of the system for conventional problems. The user would only have to exploit the systems capabilities and its complexity for new problems and exotic configurations.

4.2 Calculation Control

Not only component modelling, but also calculation processes could be described by dynamically adaptable classes. This would provide for a flexible synthesis control as well as allow adaptive analysis procedures. Moreover, taking advantage of modern operating systems, this feature could be used for using thread capabilities and thus making use of

gether with the component description, the encapsulation of the calculation procesess would represent a powerfull framework, which could be the basis for a future design system. Additionally, the class-library could be used for the development of specialized analysis tools for specific areas related with the technical and economical assessment of aircraft.

4.3 Key Problems

The proposed design system architecture, more flexible and problem-adaptive than conventional systems, nevertheless requires a thorough and detailed design. Several problems arise due to the flexibility of the system. Starting from the geometric initialization of the parametrized components, the objects themselves have to initiate the visualization of their shape as well as the widgets for graphical interactive manipulation. The storage of aircraft components with dynamically loaded cross-section modelling requires the persistance of entire objects rather than their parameters[7], which can inherently not be foreseen. If geometric parameters are to be manipulated by an optimization or synthesis process, the communication of their existance to the numerical optimizer has to be established in a dynamical way.

A further field of investigation is identified in the preparation of analysis methods for application in the proposed environment. Standard methods have to be checked for their applicability to the new architecture, especially with regard to data access and component assignment. While a detailed breakdown to bottom-level components shall be possible, the global view which is required by rough estimation methods shall be possible, too. The proposed combination of component description, container classes and virtual objects shall give this flexibility.

Finally, the development of highly comfortable and adaptable *GUI* is a major challenge for the project. The *GUI* shall hide most of the complexity from the user and implement significant parts of the design system architecture, as its workbench character is best been reflected through the interface the user is exposed to.

5. Concluding Remarks

The need for a new aircraft design system architecture is evident when studying the experiences which have been made in the last years with systems based on conventional methodologies. New respectively current software technologies promise to give a decisive clue to an enhanced flexibility with respect to emerging technologies and integration in a heterogenous concurrent design environment. The experience which has been gained on synthesis and optimization strategies with the current design systems forms a solid basis which can now be combined with software technologies, which have proven succesful in the general software world.

Object orientation, networking capabilities and dynamic adaption of data structures are seen as key elements for the realization of the envisaged design system. Not only the computing capabilities, also the programming effort during development, the quality and reliability of the system and its maintainability and extensibility are regarded as major fields of improvement. After nearly two decades of intensive research work in the field of numerical aircraft design it is necessary to synthesize the aquired knowledge and create a system, which aims for practical application in industry and research, but nevertheless offers open interfaces for further development and investigation in current and future fields of interest. Based on the experience which has been gained at *TU-Berlin* with the *CAPDA* system and its successor *VisualCAPDA*, the new approach shall be developed using the existing systems as a testbed for instant evaluation of the developed techniques and procedures.

Bibliography

(1) Ch. Haberland, O. Kranz, R. Stoer. *Impact of Operational and Environmental Aspects on Commercial Aircraft Design.* ICAS-94-1.3.1, Anaheim, 1994.

(2) Ch. Haberland, M. Kokorniak, A. Schneegans, X. Xie, O. Kranz. *Application of an Aircraft Design Tool for the Conceptual Design of Turboprop Transport Aircraft.* ISASTI-96, Jakarta, 1996.

(3) Alan H. Bond, Richard J. Ricci. *Cooperation in Aircraft Design.* MIT-JSME Workshop Proceedings, Cambridge, MA, 1989.

(4) J. S. Saggu. *Distributed Artificial Intelligence Applied to Design of Aircraft Fuselage and Wings.* ICAS-94-1.5.1, Anaheim, 1994.

(5) Ilan Kroo, Steve Altus, Robert Braun, Peter Gage, Ian Sobieski. *Multidisciplinary Optimization Methods for Aircraft Preliminary Design.* AIAA 94-4325, Florida, 1994.

(6) M. Bloor, J. Michael. *Efficient Parametrization of Generic Aircraft Geometry.* Journal of Aircraft, Washington DC, VA, number 6, 1995.

(7) S. Jayaram, A. Myklebust. *Automatic generation of geometry interfaces between applications programs and CADCAM systems.* computer-aided design, volume 22, number 1, january & february 1990.