

# THE AERODYNAMIC ASSISTANT A TOOL FOR V/STOL AIRCRAFT CONCEPTUAL DESIGN

D. J. Paisley †

*Boeing Helicopters*

J. R. Blystone ‡, G. R. Wichmann ‡, G. W. Saul ‡

*Boeing Computer Services*

Philadelphia, Pennsylvania, USA

## **Abstract**

The preliminary design of aircraft is a process that requires many iterations to develop a satisfactory solution given the constraints of the design requirements. As in any design function, creativity is a key element in the success of a design, but all too often the designer spends more time performing administrative chores than using engineering talents. As a result, the time required to develop a design using traditional methods precludes the examination of many alternatives, limiting the potential for achieving an optimal design. The design process has been facilitated by the emergence of better computing hardware, and existing software helps the designer, but does not take full advantage of advancing technologies to optimize productivity. The objective of the Aerodynamic Assistant project is to provide a suitable framework for conceptual development of an aircraft design with provision for concurrent multi-disciplinary analysis of the most up-to-date configuration, using as much of the available computing technologies as possible.

## **Introduction**

One problem when discussing the design process is the lack of a consensus for any one definition. A consistent definition is necessary to create a framework for the topics presented here. Therefore we will use one described in reference 1 and shown in Figure 1. It describes the traditional design process as we understand it, the terminology used, activities at each stage of the design and the level of detail to which a design is defined.

## **The Traditional Design Process**

The idealized view of the design process in Figure 1 shows what happens at each stage for a typical project that reaches full scale development. The following paragraphs describe the different phases in more detail.

## **Conceptual Design**

This is where many candidate concepts which may be able to satisfy the requirements are examined, and the most

promising one(s) selected for further study. The methods used to analyze the configurations are generally empirical in nature. Aircraft sizing programs are typically used to synthesize the designs. Depending upon the scope of the project, this phase may range from a few days to several weeks or months.

## **Preliminary Design**

At this point the design configuration(s) are evaluated more rigorously, using more detailed analysis tools and, where the level of effort warrants, the use of experimental facilities such as wind tunnels. This phase will usually take several months and the latter part of the effort will involve a team of designers working to define all of the aircraft structure and systems.

## **Detailed Design**

During this phase all of the design elements are refined in order to be able to produce a complete set of engineering drawings, at which point construction starts on experimental, prototype or development aircraft. One or more vehicles is produced to test the type against the customer's requirements and to identify and rectify any shortcomings of the design before full scale production begins.

## **Current Limitations**

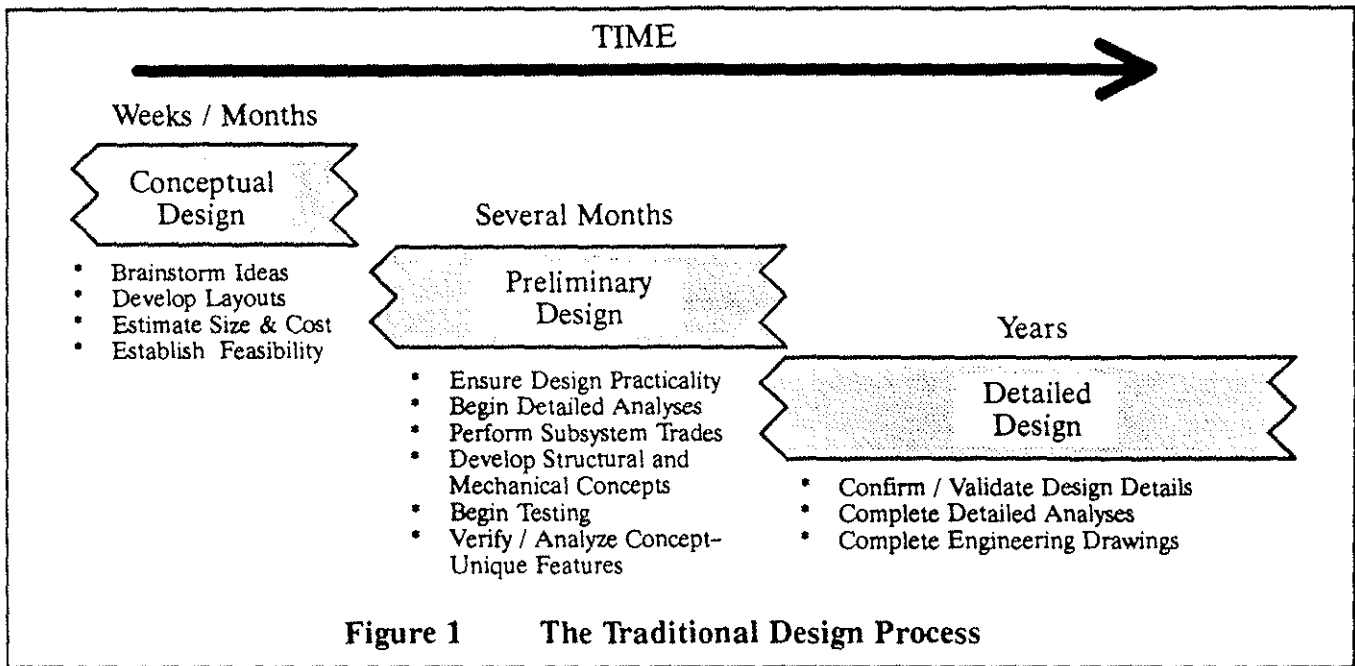
While Figure 1 represents the ideal way the process would work, Figure 2 shows a more practical picture. The execution of the process has several limitations which are discussed below.

## **Methodology**

Figure 2 shows a less theoretical model of the design process. Methods used to determine aircraft characteristics at the conceptual design stage are usually empirically based and incorporate a history of lessons learned from previous projects. A problem with empirical methods is that they are based on a limited amount of data and apply only within certain limits. Use of these methods outside of their realm of applicability should only be done with extreme caution, with a healthy skepticism for the results. The rotorcraft community is well known for producing innovative designs which when built were on

---

† *Senior Technical Specialist, Aerodynamics*  
‡ *Scientific Analyst, Scientific Systems*  
‡ *AI Specialist, Advanced Computing Technologies*



the boundaries of the current empirical databases. Some notable examples that have actually been constructed are the Sikorsky X-wing and Advancing Blade Concept (ABC) helicopters, the XV-3, XV-15 and V-22 tiltrotors, the Canadair CL-84 and Curtiss-Wright X-19 tiltwings. Key technologies for other concepts such as folding tiltrotors have been investigated both analytically and experimentally. For these concepts the traditional route of incremental development of designs does not apply very well, if at all. As a result the areas where the most assistance is needed actually have the least amount of historical data available to draw on.

Traditional design procedures would initially require developing the geometry, aerodynamics, weight and propulsion with perhaps a stab at stability and control. Substantial development of the design would take place before other disciplines, such as stress, dynamics, supportability, survivability, would see the configuration. In some cases this is because the traditional approach did not include these disciplines in the early phases of the design, such as supportability, and in other cases it is because some design disciplines require a very detailed definition of the vehicle before being able to perform meaningful analysis. An example of that is aeroelastics, where mass and stiffness distributions must be known before anything significant can be calculated. Whatever the reason, omitting key disciplines in the early design stages can lead to situations where poor design features are locked into a design before the engineer responsible for that discipline is even aware of the project. The end result is that the feature is either kept and a penalty paid for poor design, or the problem is fixed, usually at great cost.

### Existing Analysis Software

A good deal of existing engineering software dates back to the 60s and 70s, when machine hardware and compilers were much less capable than they are now. A result of this was that old codes were frequently 'shoehorned' into the available space, with sacrifices made to clarity and maintainability of the code. As these codes have been heavily used and extensively modified over the years, the problems have grown with the code. Software Engineering techniques make it plain that it is cost effective to rewrite software from the ground up after about five years of such a modify / use cycle. The cost of not rewriting the code is hidden in the length of time it takes to make enhancements and in the debugging of such enhancements that are made to the old code. Or worse, needed enhancements are not implemented because of the fear of interfering with some other working portion of the code.

### Data Entry and Program Usage

A major problem with current sizing and analysis methods is the need for the engineer to keep track of many hundreds of parameters manually. A typical sizing program is written in FORTRAN and the input consists of several hundred numbers. Admittedly, some of these numbers fall into major data groupings, such as an engine deck, which are not usually entered individually. Nevertheless, there remain hundreds of single parameters which must be kept up to date and consistent.

User interface techniques available to programmers for data input in the 60s and 70s were limited to punched cards and teletype terminals, with line editors the primary on-line means of modifying data. Many of the programs developed for that environment are still functioning in

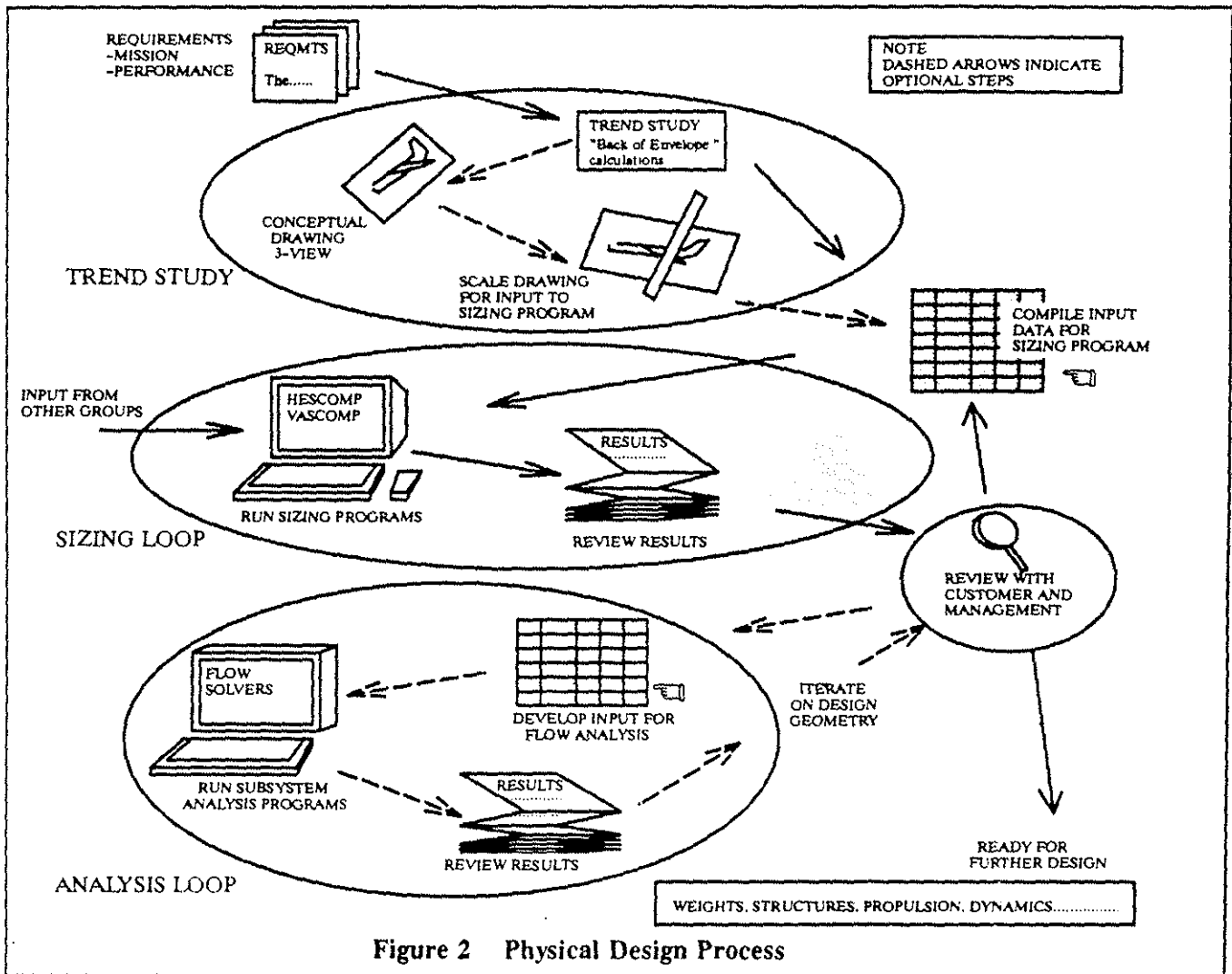


Figure 2 Physical Design Process

that mode today. Some recent programs use the namelist input technique, but even this relies heavily on cryptic FORTRAN variable names, typically squeezed into no more than 6 characters (ISO Standard FORTRAN) to describe the input data. The cryptic nature of the input process leaves the user prone to mistakes and makes the task of training new people a more lengthy process than it need be.

The evolution of a design requires many runs of these sizing programs, and it is up to the user to manually keep track of the evolution of the input data which results in the final sized aircraft configuration. This is a time consuming administrative burden on the engineer, which reduces his or her productivity.

### Lessons Learned

As a design develops, the engineer learns many things about the particular configuration being developed. Some of these insights are relevant to all or most configurations, some only to similar configurations. The dissemination of these insights is currently up to the individual engineer.

If the engineer elects not to share the information, then much duplication of effort results, due to other engineers having to learn the same lesson themselves. On the other hand, if everyone tries to disseminate everything, there is a danger of useful information being lost in the landslide of paper.

### The Future of the Design Process

Figure 1 implies that detailed design is the most important part of the process as it takes the longest. Studies show, however, that in the development of a new aircraft more than 80% of the cost of the vehicle is locked in by the time 5% of the money has been spent. The implications of this are that money spent in the conceptual and preliminary design stages to produce a better (i.e. lower cost) design is money well spent. It is also true that for every project which survives to at least the flying prototype stage there are hundreds of projects which do not, in addition to studies which are never intended to go beyond the "paper airplane" phase. Thus the amount of time that design and

technology groups spend doing conceptual design is much greater than is implied by figure 1.

The future of the design process can be shaped dramatically by the tools now becoming available. These tools will greatly aid the engineer in:

1. Capturing Requirements Better
2. Improving Quality of Design
3. Reducing Design Cycle Time
4. Saving Lessons Learned

### Design Requirements

The very first phase of design is really a pre-design phase, that of requirements gathering. There is a school of thought that determines that design begins when someone draws something, a concept which obscures the fact that some design requirements had to exist before the designer began to draw. Requirements do not always come as several pounds of documents, thankfully. Requirements can range from a germ of an idea to a full blown government specification, with all that entails. It is important to recognize that however well hidden it may sometimes be, the requirements phase does exist and recognizing that fact it usually pays to be intentional about gathering as much in the way of pertinent requirements as possible. Depending upon the scope of the project this phase may take several days or merely a few hours.

### Design Quality

In a manufacturing environment quality can be defined as being able to repeat a task exactly when required to do so. This results in the production of identical parts which eliminate scrap and minimize cost. Thus consistency and predictability are synonymous with quality when producing a product. Although the design process is creative and innovative, it too can benefit from employing consistent procedures and methodologies. One way to achieve this is to give the designer a tool or set of tools which facilitates these goals. ~~this~~

The quality of the design process can be improved by using better analysis techniques from a wider range of engineering disciplines earlier in the sizing. This will help to identify characteristics more accurately and to find unacceptable design qualities at a stage where they can be more easily remedied, instead of being hidden behind the inadequacies of empirical methods. This process would apply to any design process, but the complexity of V/STOL designs and the inter-relationship of so many technical disciplines make the potential payoff so much greater.

As an example, Reference 2 describes the ADDSS/EDRAS system which allows the designer to develop systems at the preliminary design level, with guidance available on-line to evaluate a wide range of relevant criteria. The initial focus of the project was to integrate supportability issues into the design loop.

### Design Cycle Time

Improving the designer's access to analysis capability and removing most of the administrative burden will significantly reduce the amount of time required to perform the basic tasks. A further benefit is that a greater proportion of the designer's time will be spent doing productive work.

### Design Experience

Capturing and storing design experience in terms of "lessons learned" as the design progresses will allow other engineers to benefit from any insights other designers have had on related topics. A system which can remove the burden of disseminating the information, but provide it in the right context will help greatly with sharing experience between designers and improving the training of new people.

### The Aerodynamic Assistant

The Aerodynamic Assistant project was initiated to address most of the problems described above, and to provide a framework for future enhancements. The project has been developed using the methodologies of Structured Analysis and Design for software, described in References 3 and 4. Figure 3 shows the logical flow diagram of the Aerodynamic Assistant. It can be seen that the hub of the whole diagram is the Aircraft Configuration data store. Once a configuration has begun the parametric sizing process, the data store holds all of the relevant information for the aircraft configuration. From this core, the parametric geometry tool and various analysis packages can use this data to refine the configuration. Refinements are passed back to the datastore. These refinements then trigger re-sizing of the aircraft. The process continues in an iterative fashion until a solution has been reached.

### Hardware and Software Standards

In order to remain hardware and software independent, and in order to write as little new code as possible, the Aerodynamic Assistant was designed to take advantage of software and hardware standards wherever possible. The emergence of the X Window System and the products layered upon it, for example OSF/Motif, made it a clear choice for implementing a portable graphical user interface. Relational database systems provide a way of storing the large amounts of data required for the system. Finally, standards for 3-D graphics such as PHIGS enable portable geometric tools to be developed.

### Graphical user interface

The goal of any user interface is to make a software program intuitive and easy to use by a novice. In addition the Aerodynamic Assistant user interface allows the designer to focus on the engineering problems and does

not force him or her to become a computer super user in order to use it effectively.

The user interface for the Aerodynamic Assistant is designed with three goals in mind:

1. Provide a consistent interface across all tasks
2. Provide engineers who have varying levels of expertise with the support necessary to successfully accomplish their tasks
3. Provide an event driven interface which follows rather than leads the engineer in the execution of tasks.

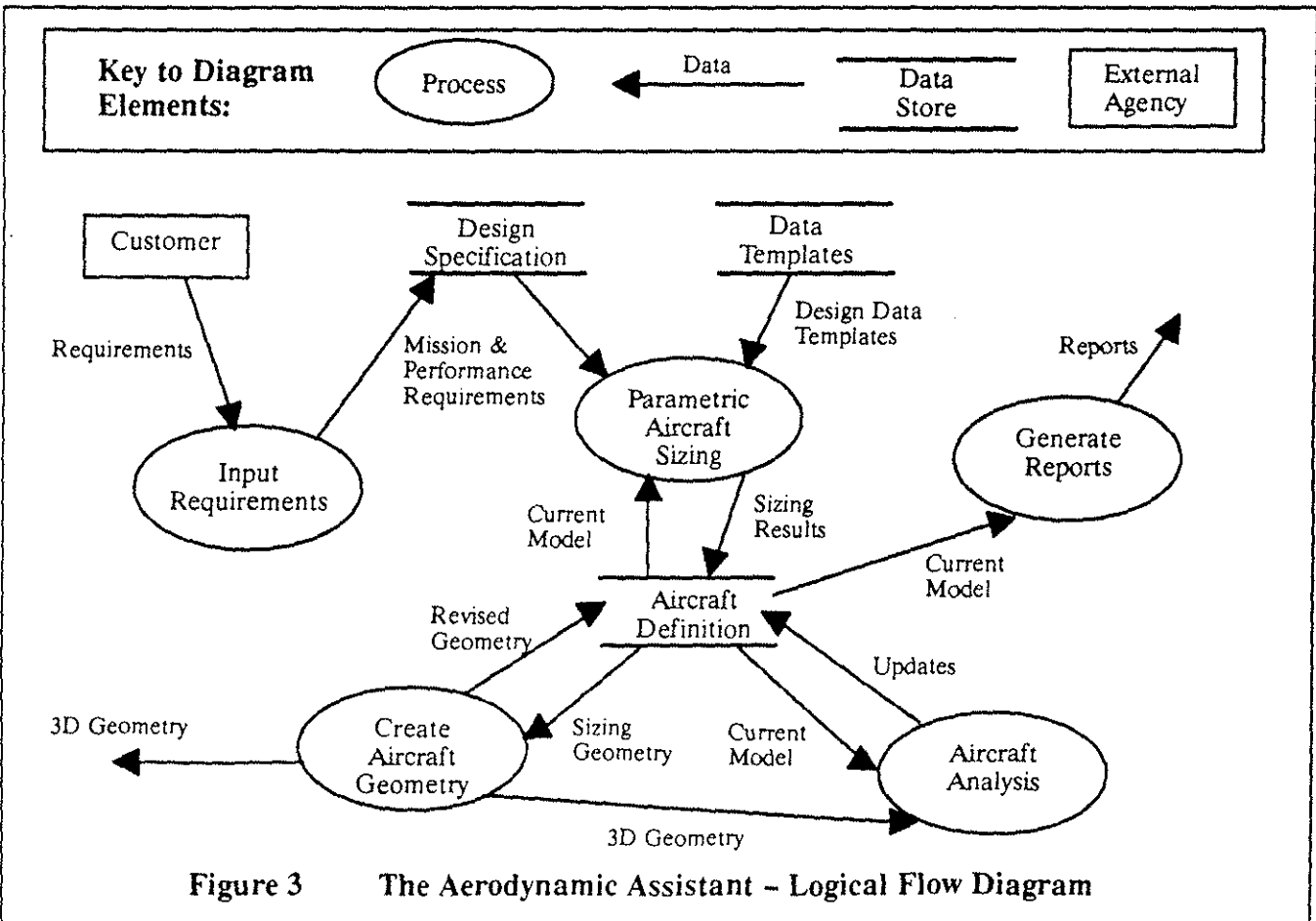
Figure 4 gives an indication of how the user would see the user would see and use the system. The user interface is hierarchically structured, with a system level, a project level, and a model level. Each level provides a window with a menu bar which allows the engineer to perform tasks relevant to that level. In a typical scenario, the engineer will create a project using the menus in the system level window, create a model using the menus in the project level window, then develop the model using the menus in the model level window. The windows themselves are created as individual processes. The engineer may use this feature to display and work on several projects and/or models at the same time. The

primary task of the system and project levels is to provide the engineer with the capability to manage projects and models. In addition, the project level facilitates the input of the design requirements, and the execution of the trend study.

The model level provides access to all of the design tools. At this level, the engineer performs an iterative preliminary design loop which involves the execution of the sizing code and the geometry management tool. The capability to transfer a result of this loop into detailed analysis codes is also supported at this level.

At all levels, online help is provided to assist the engineer both in running the Aerodynamic Assistant, and by providing information relevant to the design process. The primary use of this second type of help will be to provide guidance to new engineers.

Other features of the user interface include the minimization of input through the use of both system and user defaults, pop-up windows to provide warnings or other pertinent information, bounds checking to help screen input errors, global units definition to permit a straightforward change of unit systems (i.e. from metric to english), and the use of color in a consistent manner to logically group relevant information.



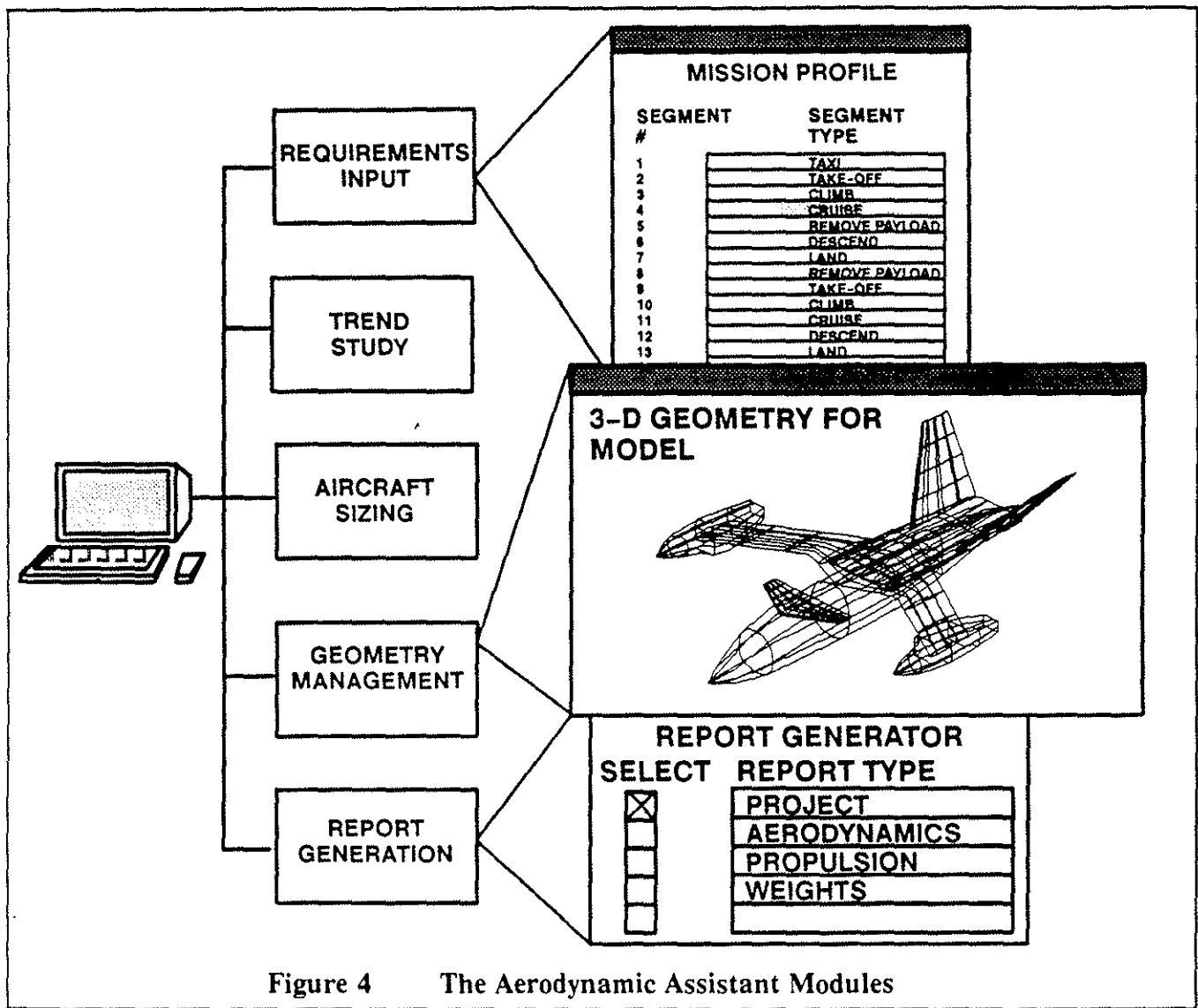


Figure 4 The Aerodynamic Assistant Modules

### Data Base Management Systems

Databases form the core of the Aerodynamic Assistant and provide the foundation for the overall integration of the system components. Generally, databases provide the management of persistent data and the ability to access large amounts of data efficiently. Each engineering database can be the application data bridge to CAE (Computer Aided Engineering) environments of other design disciplines in aircraft vehicle technology. Distributed databases will provide the foundation of automations efforts, such as integrated engineering and concurrent engineering. The Aerodynamic Assistant system is a building block toward these efforts. A database management system (DBMS) will be used to store the data for each aircraft design. The implementation of a DBMS will allow all engineering groups associated with the preliminary design to refine the design and coordinate activities early in the conceptual stage.

The database for the Aerodynamic Assistant requires the ability to store a variety of data types: textual, mathematical (numerical), graphical (2-dimensional), and geometrical (3-dimensional). For a more detailed description of these data types see Reference 5. References 6 and 7 also discuss the requirements of engineering oriented databases.

The databases in the Aerodynamic Assistant are logically group into two types, Design Databases and Concept Template Databases.

### Design Databases

The Design Databases contain data specific to a design project, the conceptual models in that project, and the performance / design analysis for verifying and validating the conceptual design. The Design Databases will be physically separate for each project, which provides the exclusivity and security for classified design projects. The

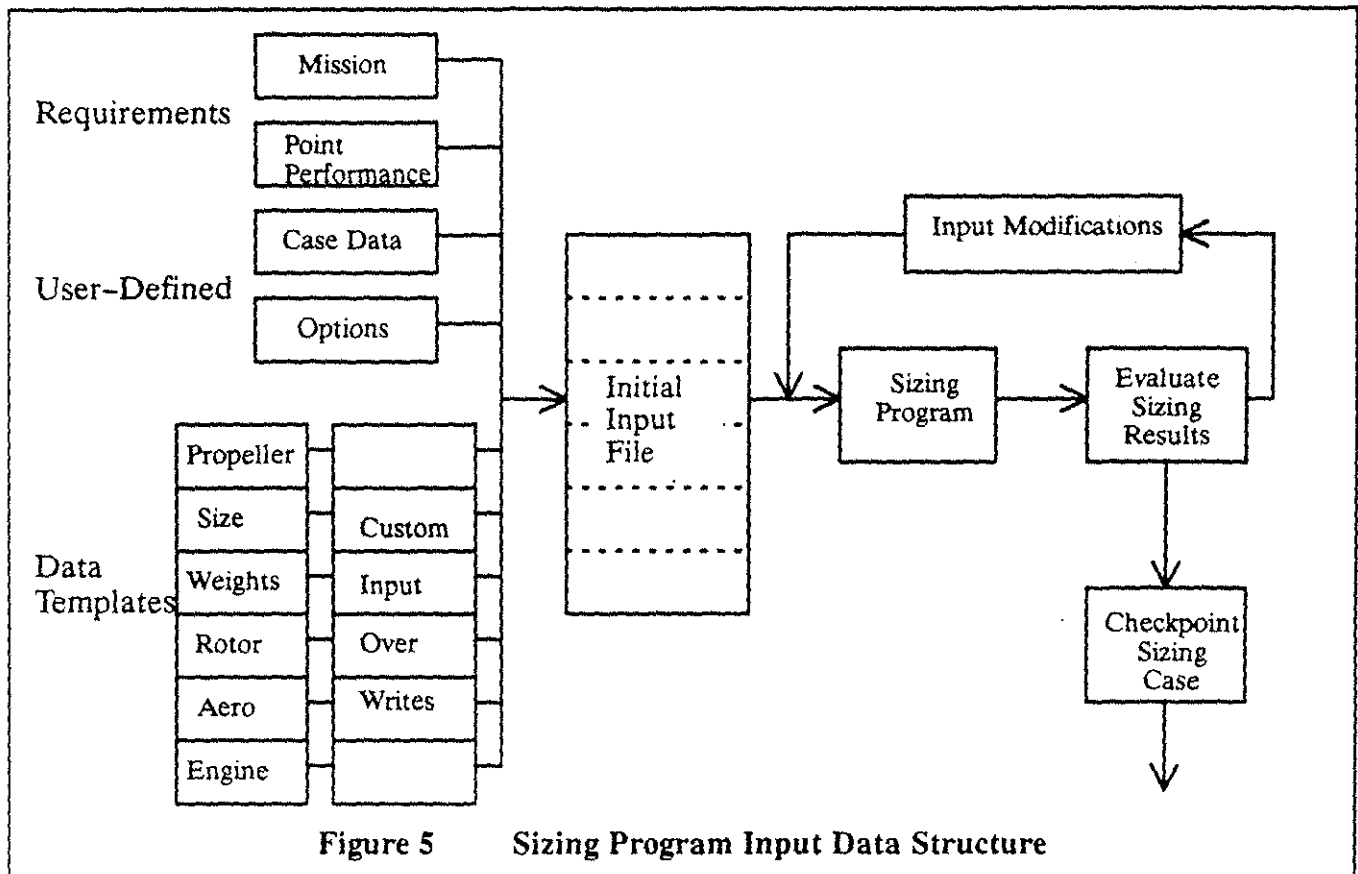


Figure 5 Sizing Program Input Data Structure

various data that are stored can be separated into the following groups:

1. **Requirements Data.**  
This captures all of the customer requirements relating to vehicle performance, for instance the mission profile, point performance targets such as dash speed and hover requirements. Future expansion would allow broader requirements to be captured, such as reliability and maintainability, using a hypertext tool to capture the data as described in Reference 2.
2. **Design Data Templates.**  
These are compilations of input data for previous designs with specific attributes, broken down by category (Weights, Aerodynamics, Geometry, Propulsion, Rotor, Sizing Options). Sizing inputs for a new project can be made up by selecting a variety of templates from past designs. The user can then overwrite the values contained in the templates to more accurately reflect the requirements of the new design.
3. **Aircraft Design Data.**  
This includes all of the data necessary to describe the vehicle characteristics, performance and geometry at the conceptual design level. The geometry would be specified to at least the level required to define a

typical three view drawing, with the capability to develop it further in order to be able to generate panel models and, if desired, CFD grids.

4. **Aircraft Performance Data.**  
This data includes mission performance data, point performance data and more general data such as flight envelopes, maneuvering performance and payload - range/radius.
5. **Project Log.**  
The project log is a history of the project containing the information about what was done and when. It automatically logs sizing runs, the major features of the cases that were run and a comment from the user. This will provide a useful audit trail of design decisions.

### Concept Template Databases

The Concept Template Database contains information on past designs. Its modular format allows new designs to be developed from components of previous work. The components of the design are broken up logically into design categories and characteristics, as shown in Figure 5, which also shows how the data is combined and modified through the design cycle. By categorizing the design components, design templates are created which the design engineers can pick and choose from to develop a

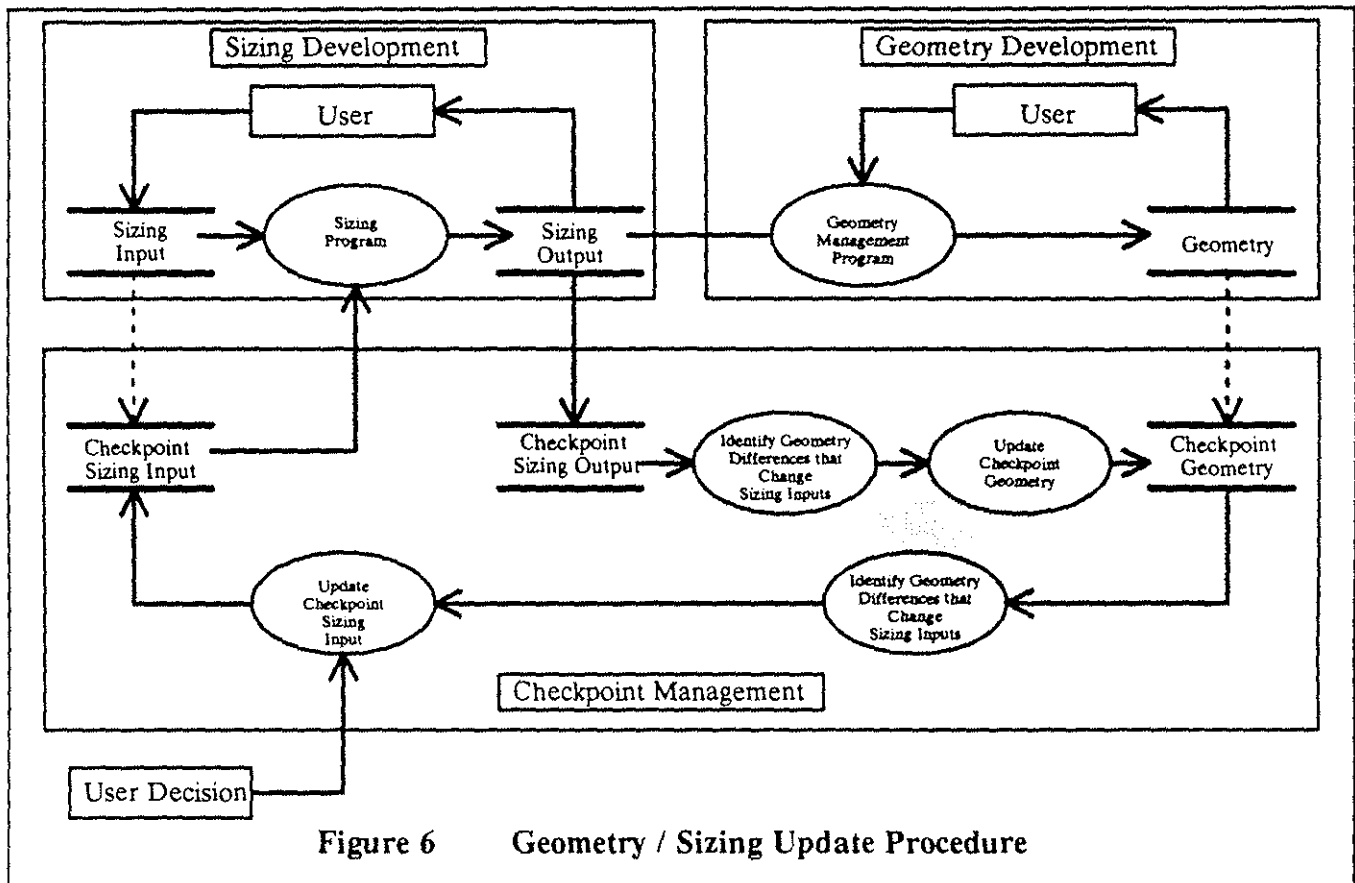


Figure 6 Geometry / Sizing Update Procedure

new design baseline. The input data for sizing / synthesis programs can be created from these templates. The typical categories for the templates in the Aerodynamic Assistant are:

1. Geometry - which values are fixed, which are variable.
2. Weights - parametric weight trend constants and multiplicative factors.
3. Aerodynamics - lift and drag characteristics.
4. Propulsion - engine decks, rotor sizing.
5. Mission profile.
6. Sizing options.
7. Rotor performance, where applicable.

The Concept Template Databases can be shared among all design projects. The templates can be referenced by design class (i.e. Tilt Rotor, Tilt Wing, Lift Fan) and by general performance / design features. This search will produce a list of past concept templates that the design engineer can select from.

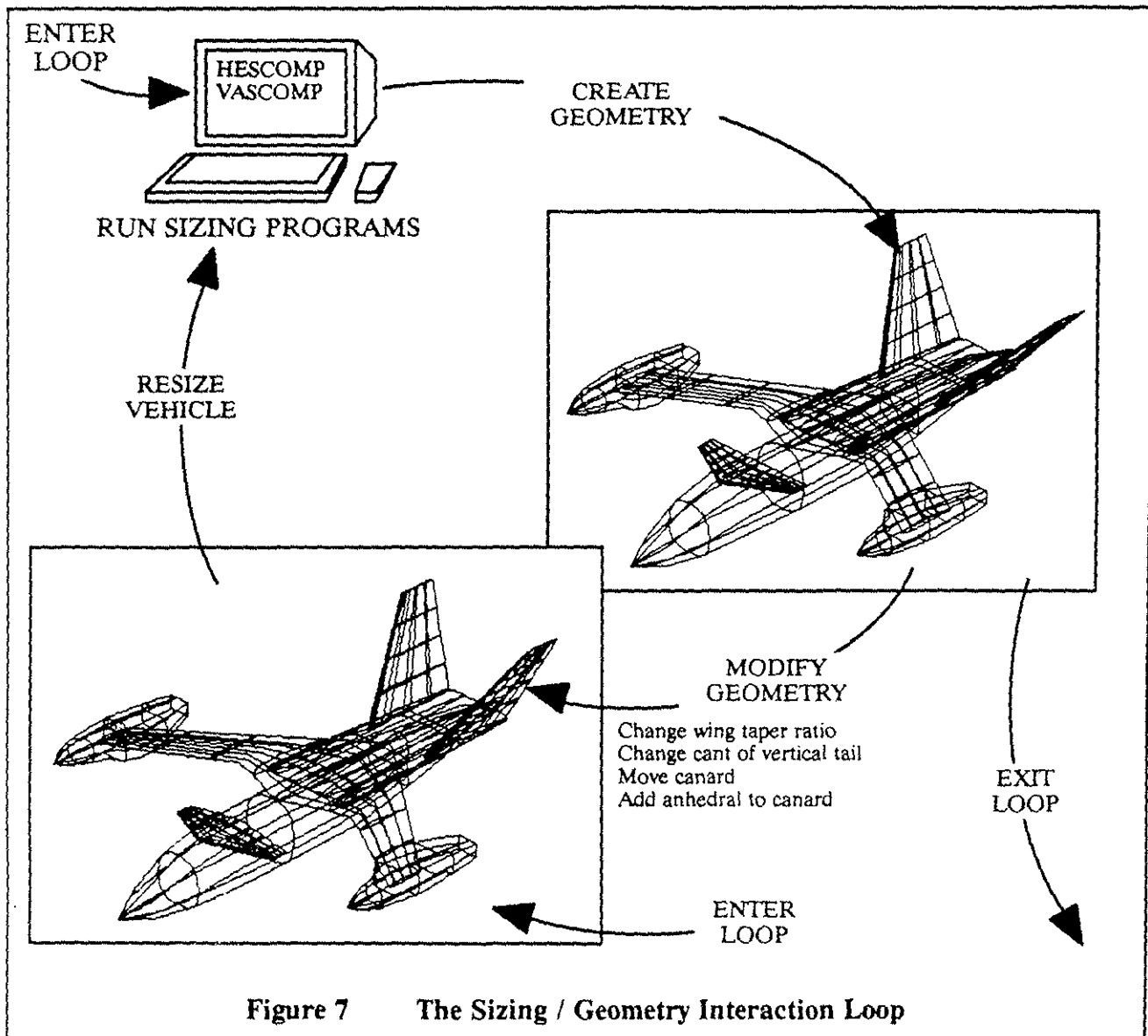
#### Integration of geometry and analysis tools

Developing three dimensional geometry from the output of a sizing program is an excellent way to provide feedback

to the designer while maintaining consistency with the current design data. Rapid visualization allows the intuitive and artistic nature of the design process to proceed more confidently. In addition it provides a framework from which many engineering disciplines can develop their own unique models. Most sizing programs provide at least the bare minimum of geometric data required to generate full three-dimensional geometry. References 8 and 9 describe methods which have been developed to create geometry directly from the output of sizing programs. The Aerodynamic Assistant uses the geometry development module from ACSYNT (Reference 9) to generate fully parametric geometry. The user can enhance and modify the geometry using the tools available in ACSYNT. This geometry is tied to the parameters generated by the sizing program, and so a modification of the geometry may require resizing of the aircraft and modification to the sizing for any reason will affect the aircraft geometry.

Synchronizing the simultaneous development of the aircraft sizing and geometry is one of the primary goals of the Aerodynamic Assistant. Only by achieving that can the design proceed in an orderly fashion. Thus it is necessary to track changes made to the design in order to ascertain whether the latest sizing run or geometry has become invalid. Figure 6 is a schematic diagram of how the Aerodynamic Assistant manages the simultaneous





**Figure 7 The Sizing / Geometry Interaction Loop**

development of geometry and sizing data. The sizing and geometry can be developed independently, but eventually the two must be integrated. The points at which this occurs are designated checkpoints, and the third major function of the update procedure is managing these checkpoints.

Figure 7 shows the process more graphically. The figure shows three major functions, sizing, geometry creation and geometry modification. The design process can begin at any point, but when working from a firm set of requirements, the sizing will usually be done first. The geometry will be created when the sizing has been developed to a suitable initial design point. The user can then modify either the sizing or geometry independently of the other.

There is a checkpoint management function which notifies the designer when either the sizing data or

geometry is invalidated by a change in the other, necessitating an update. Forcing an update is a decision which the user can then make when a suitable point in the design cycle is reached. The design cycle continues until a satisfactory conclusion is reached, at which time a complete set of aircraft configuration data is available from the sizing analysis and matching aircraft geometry is available from the geometry tool.

### Design Knowledge Bases

Reference 10 gives a discussion of the use of expert systems as aids to the aircraft design process, while Reference 11 describes a successful expert system based aircraft synthesis system. The implementation and effective use of knowledge bases require that the system is well structured and that data is easily accessible. Earlier approaches to expert systems and knowledge bases

warranted that the expert system be in complete control and function as the "expert executive" of the system. This approach tends to get in the way of the design engineer who has his own techniques and methods for addressing design problems. The Aerodynamic Assistant applies knowledge bases as utilities in difficult areas of the design cycle so that the "assistance" can be asked for just as one would go to a senior designer for help. The storage and facilitation of past knowledge (lessons learned) is vital to continued improvement. Those who cannot remember the past are doomed to repeat it. Inexperienced design engineers can learn without having to suffer the worst effects of making poor design decisions and can be more productive with their time.

The knowledge base structure should segment the domain knowledge into logical and related pieces. These "knowledge islands" will help the system to run more efficiently and simplify enhancements and maintenance to the individual knowledge bases. The knowledge bases will form an integrated set of rule and object networks used to store heuristics and attributes about design knowledge. The various knowledge bases in the Aerodynamics Assistant will:

- assist in trend study analysis
- execute the proper compiled sizing subroutines
- provide guidance in the development of the 3-view concept design drawings
- provide warnings on violating design constraints and advice on how to correct problems
- help in diagnosing problems in panel model analysis and finite element model analysis.

### Conclusions

The Aerodynamic Assistant will integrate current design tools at Boeing Helicopters, and provide a modular structure for incorporating new tools as they are developed. By using a database management system for aircraft design, the problems associated with data redundancy between disciplines and the manual transfer of data between analysis applications will be eliminated. The augmentation of individual design expertise with knowledge based expert systems will enhance the engineer's capability to produce viable designs.

The implementation of this package on a network of high-powered graphics workstations will provide the designer with a powerful personal design tool.

Once the concept has been proven and tested the Aerodynamic Assistant will provide a strong foundation for the integration of other technology disciplines into preliminary design.

### References

1. Rosenstein, H. J., "Conceptual Design of an Advanced V/STOL Special Electronic Mission Aircraft," American Helicopter Society Specialists Meeting on the Rotary Wing Aircraft Conceptual Design Process, April 1989
2. Beggs, R. M., ADDSS/EDRAS Influencing Preliminary Design For Enhanced Supportability, Proceedings of the 46th Annual American Helicopter Society Forum, Washington D.C., May 1990
3. DeMarco, T., Structured Analysis and System Specification, Yourdon Press, 1979
4. Page-Jones, M., The Practical Guide to Structured Systems Design, Second Edn., Yourdon Press, 1988
5. Smith, Mark K. and Tockey, Stephen R., "An Integrated Approach to Software Requirements Definition Using Objects," Tenth Structured Development Forum, August, 1988.
6. Fulton, R. E., and Yeh, Chao-pin, "Managing Engineering Design Information," AIAA Paper No. 88-4452, AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, September 1988.
7. Dayal, U., Heiler, S., Orenstein, J. and Radke-Sproull, S., "An Object-Oriented Approach to Data Management: Why Design Databases Need It," 24th ACM/IEEE Design Automation Conference, 1987.
8. Lu, Liang-Ju, Myklebust, A., War, S., "Integration of a Helicopter Sizing Code with a Computer-Aided Design System," CUE International Conference, May 1986.
9. Gelhausen, P., Jayaram, S., Myklebust, A., and Wampler, S.G., "Improving Aircraft Conceptual Design - A PHIGS Interactive Graphics Interface for ACSYNT," AIAA Paper No. 88-4481, AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, September 1988.
10. Eskey, M. and Kidwell, G., "Expert Systems and Their Use in Augmenting Design Optimization," AIAA Paper No. 85-3095, AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, October 1985.
11. Kroo, I., and Takai M., "A Quasi-Procedural, Knowledge-Based System for Aircraft Design", AIAA Paper No. 88-4428, AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, September 1988