'91 KACC 1991. 10. 22~24

A CAI System for Conceptual Design of Aircraft

Yoshisada Murotsu, Showzow Tsujio and Choong Sik Park

Department of Aeronautical Engineering
University of Osaka Prefecture, Sakai, Osaka 591, JAPAN

Abstract

A CAI system is developed to support the instruction of an aircraft conceptual design for aeronautical engineering students. Three system concepts are proposed and an Object-Oriented approach is applied to construct the system. The system has three major functions to perform a conceptual design: (1) the system stores modular data and empirical formulas used for a wide range of aircraft design tasks from light aircraft to long range airliners. (2) Implementation of modules by message passing makes it easy to realize the various design tasks required for different design requirements. (3) The system allows users to study trade-off among the requirements. The system has a graphical user interface which allows users to communicate with the system interactively. The effectiveness of the system is demonstrated through some case studies.

1. Introduction

It is widely perceived that design education for aeronautical engineering students is in a state of crisis[1]. There are two reasons for this: (1) Most aeronautical engineering departments do not have sufficient knowledge and data about aircraft design. (2) No standard methodologies of aircraft design exist, especially of conceptual design. There is little possibility of establishing the methodologies, because the complexity of aircraft design obliges designers to take trial and error processes.

In other fields of aircraft design, computers are applied to process design information efficiently. In fact, many programs are employed in various fields of aircraft design, i.e. database systems, CFD(Computational Fluid Dynamics), CAD, structural analysis programs and so on. Unfortunately, there are few programs for an aircraft conceptual design. The cause of this is that procedural programs are not appropriate for describing the data and design tasks which have the characteristics as follows[2,3,4]:

(1) Algorithmic procedures cannot be defined.
(2) There are various kinds of design data, i.e. numerical values, symbols, graphs, and drawings.
(3) There are various ways to process data.
(4) Data and empirical formulas utilized for a design are often updated.

The purpose of this study is to propose the realization concepts of the CAI system for the conceptual aircraft design and to implement the system using an Object-Oriented approach.

This paper consists of six sections. The rest of this paper is organized as follows. Section 2 discusses aspects and features of an aircraft conceptual design. The system configuration is described in Section 3. The new concepts are proposed for the development of the system, using an Object-Oriented approach. In section 4, design process is represented by modules and message passing. Section 5 presents several case studies using the developed system and discusses the effectiveness of the system. In the final section, some concluding remarks are given.

2. Aspects of conceptual aircraft design

The aim of conceptual aircraft design is to determine basic specifications of an aircraft which satisfies design requirements.

Table 1 shows a typical example of design requirements while Table 2 does the resulting design specifications. The conceptual aircraft design process[5] consists of many steps as illustrated in Fig.1. The design steps are briefly summarized as follows:

(1) Design requirements

Design requirements are proposed by customers. Customers specify performances and functions of an aircraft from its mission requirements. The mission requirements are defined as a number of sectors listed for the purpose of the aircraft. Different missions result in different design procedures. In this stage, designers will set some specifications satisfying the constraints of the design requirements, for example, a stall speed and a take-off length.

(2) Wing Geometry

Designers set some of principal dimensions of wings and tail wings. Basic design variables are airfoil shapes, an aspect ratio, a taper ratio and a sweep back angle.

(3) Thrust-to-weight ratio and wing loading

Thrust-to-weight ratio and wing loading are key design variables which are estimated in all the mission sectors, referring to tables and formulas derived from existing data.

(4) Initial sizing

The process of initial sizing determines a take-off weight by the use of an approximate formula.

(5) Layout data

Layout data are checked from three views of an aircraft.

(6) Aerodynamic characteristics

Table 1 Design requirements

Design requirements
Range Number of passengers Payload weight Maximum speed Takeoff length Cruise speed Cruise altitude Stall speed Rate of climb

Table 2 Design specifications

Design specifications				
Performance	Geometry			Others
	Fuselage	Wing	Tailwing	
the same as design requirememnts + the others	length diameter	area span chord (root,tip) sweep-back-angle	the same as wing	position weight MAC C.G

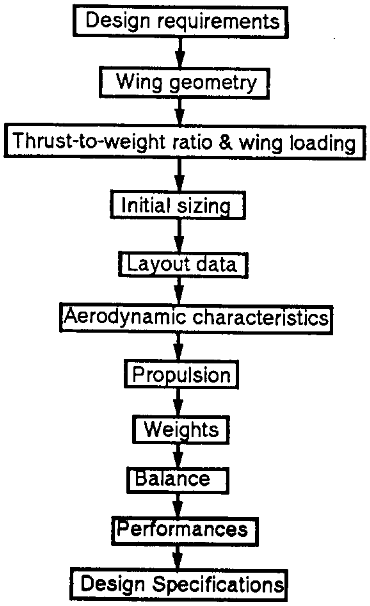


Fig. 1 Design process

Aerodynamic characteristics of the aircraft are evaluated in terms of a maximum lift coefficient, a lift curve slope, a parasite drag coefficient and an induced drag by using approximate formulas.

(7) Propulsion

Performances and dimensions of propulsion systems are checked by using the plots of specific points of existing data.

(8) Weights

The weights of aircraft components are approximately calculated by using empirical formulas which are functions of a takeoff weight and some other design variables. For example, a wing weight is determined from a take-off weight, an aspect ratio, a wing area, a taper ratio, a thickness ratio, a sweep back angle and a load factor.

(9) Balance

The balance of an aircraft is checked by the locations of center of gravity under specified flight conditions.

(10) Performance

The performance of an aircraft is evaluated by using design variables determined in the above design steps and checked with the design requirements.

As stated above, the characteristics of the aircraft conceptual design are to determine kinds of design variables and their values by using data and empirical formulas. Especially, different aircraft types are designed by using different design variables and procedures. The design processes are complex and iterative, so that designers must satisfy all the design requirements by trial and error.

3. Configuration of the CAI system

3.1 System concepts

An information processing system is required to represent existing design data and design tasks in order to apply them to various design requirements. The system has been developed, based on three concepts which are outlined below.

(1) Data structures

The data in the system is required to have structures because designers do not treat data as numerical and symbolic values, but as structures. For example, a designer may divide an aircraft design into four aircraft part designs, i.e. a fuselage design, a wing design, a tail wing design and a propulsion design. Hence he treats aircraft data as data structures with respect to the four parts.

Furthermore, a designer deals with a design object from multiple view points. In wing design, he may deal with geometric data which consist of a wing area, a taper ratio, an aspect ratio and a sweep back angle. Next he will treat aerodynamic coefficients for determination of aerodynamic characteristics.

Figure 2 shows a structure of aircraft data. The data structures of the system has to be composed and decomposed during design process. A large number of data must be retrieved effectively.

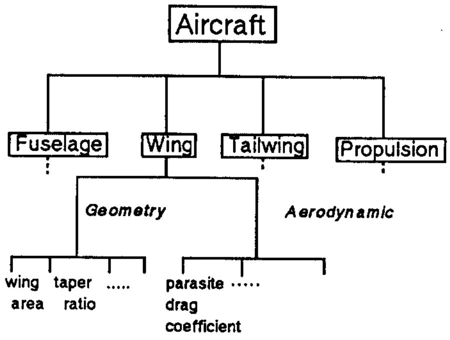


Fig. 2 Data structures

(2) Modules

In conceptual design, data and design tasks are not definite. Therefore, the system need be composed of modules which perform an independent function. In case of dealing with a canard, he considers canard geometric variables, aerodynamic coefficients and weights. These design variables are closely connected with each other, and then the design processes are necessarily proceeded by "trial and error" ways. In order to support a trial and error process, the system is required to consist of modules, which are effective for supporting the try and error processes. The supporting functions are presented in the following operations:

(a) Operation of a design variable, i.e. input and output of a design variable's value, creation and deletion of a design variable.

(b) Operation of a database, i.e. addition, deletion and retrieval of data.

(c) Operation of a design task, i.e. implementation and modification of a design task.

(3) Readable programs

The system is used as a CAI system. Hence the system is preferable to be described in readable manner. Such readable programs are also valuable as manuals.

3.2 Object-Oriented approach

An Object-Oriented approach[6] is employed to develop the CAI system based on the three concepts mentioned above. The concepts are realized by an Object-Oriented approach.

(1) Object-Oriented data structures

An object consists of instance variables (specific data) and methods (specific procedures to operate instance variables). The hierarchical class-system makes complicated data structures simple.

(2) Realization of modules

A message expression describes a minimum size module which operates design variables. A collection of modules describes a module of design task. When data or an operation are changed, only a corresponding message expression is changed.

(3) Description of readable programs

In case of dealing with a design process as objects, readable programs can be successfully constructed by using classes, instance variables and methods.

3.3 System configuration

The CAI system is developed with Sun spark work station and Smalltalk-80 version 2.5[7]. Figure 3 shows the configuration of the CAI system. The modules communicate with each other by sending messages(message passing). A solid arrow line depicts message passing. A broken arrow line denotes data transfer.

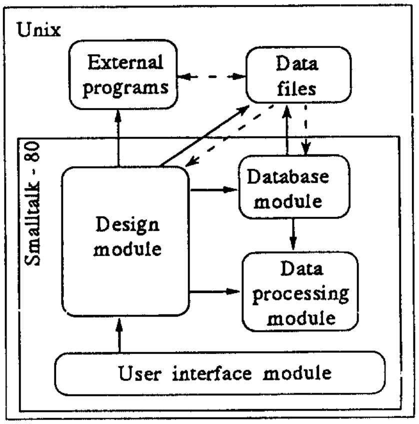


Fig. 3 System configuration

Functions of each module are described in the following.

(1) Design module

The design module is realized by using classes. A class possesses a database which stores existing aircraft data as one of instance variables. The class also defines methods as design procedures which represent design steps and design tasks. During execution of the system, users select methods corresponding to the design tasks if they require them. If a set of methods to be performed is defined before start of a design, the system sequentially evaluates them.

(2) Database module

The database has two functions:

(a) creation of data

By initiating a database module, the module reads data from data files of Unix, and then objects of data are generated. The system creates, deletes and modifies the objects during the execution of design tasks.

(b) operation of database

The operation of database is defined by using methods. Users can retrieve the data by using composite message expressions under complicated conditions of design tasks.

(3) Data processing module

The data processing module plots specific data points on a two dimensional graph. The module provides various methods for graph operations, i.e. interpolation, extrapolation, change of displayed attributes, and change of an axis scale. Users can get and retrieve data by selecting a corresponding point by a mouse.

(4) User interface module

The user interface module contains basic methods to define displayed layout and items of menu. Users construct their favorable user interfaces by modifying only a few parts of the methods.

(5) External programs

The external programs can be written by the C language to compensate Smalltalk-80's numerical calculation ability. The kernel of the initial sizing module is programmed as an external program.

(6) Data files

The data files store the data described below.

- (a) existing aircraft data(performance, size, three views).
- (b) statistical graph data (sweep back angle versus aspect ratio, take-off parameter versus take-off distance, wetted aspect ratio versus maximum lift to drag ratio).
- (c) Input and output data between the system and external programs.

4. Description of design process

Class DesignStepObject and class DesignTaskObject describe the conceptual design process of aircraft, which are outlined as follows:

(1) DesignStepObject

An instance of the class has two design informations: (a) names of design steps and (b) names of design task of each design step.

(2) DesignTaskObject

The class defines methods as procedures of design tasks. Figure 4 shows a method of a design task to set a wing span.

```
calculateSpan                                      ----------------------(a)

   | aSpan aWingLoading anAspectRatio|  --------------- (b)

 aSpan <- designVaribleDictionary                          ┐
                           select: 'Span'.                 │
aWingArea <- designVaribleDictionary                       │
                           select: 'Wing Area'.            ├--- (c)
 anAspectRatio <- designVaribleDictionary                  │
                           select: 'Aspect Ratio'.         ┘
```

Fig. 4 A method of a design task

The method is described in the following format.

- (a) description of name of method
  The name corresponds to a name of step.
- (b) description of local variables
  Local variables used in a method are described.
- (c) calling of design variable
  The message expression is to call a design variable from an object(designVariableDictionary) which store all of the design variables used in the design. If a design variable is not stored in it, the system asks a user whether the selected design variable is allowed to be stored.
  Therefore, the system deals with change or refinement of design during a design process.
- (d) operation of a design variable
  A design variable is an object with its own name and value. Sending a message "value" refers to its value.

Figure 5 shows a process of performing a set of design tasks by message passing. In message passing, a method is evaluated by matching the name of its own method to a symbol as a sent message. At first, a message sent by a user is matched to a method described in DesignStepObject, and the method is evaluated. Next, a message (perform:designTask) is sent to an instance of DesignTaskObject(designTaskObject). Accordingly, a value (setMaximumSpeed) is set to an instance variable(designTask) included in an instance of DesignStepObject. In the similar way, a method (setMaximumSpeed) is evaluated.

Such a programming paradigm which consists of modules and message passing provides easy description of a complicated design process. Change or refinement of design tasks is replaced by corresponding message passing or methods. Moreover, modules with suitable names can provide readable property to the programs, so that the programs become valuable manuals.

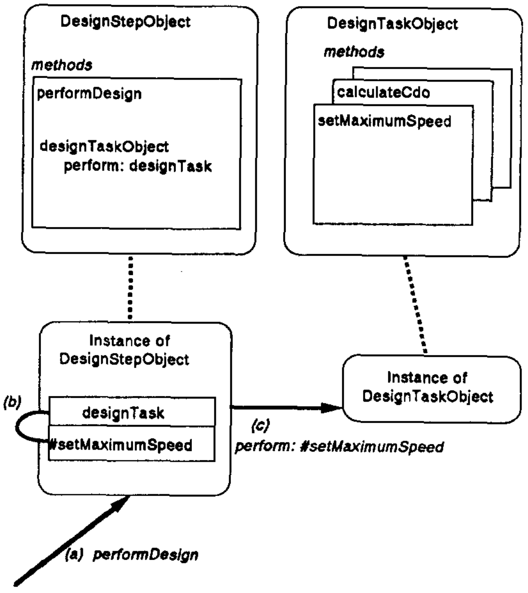
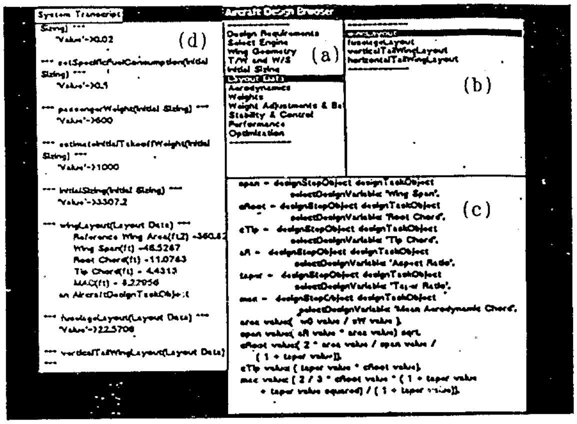


Fig. 5 Process of message passing

5. Numerical examples

Three case studies are presented by using the developed systems. In the first case study, a conceptual aircraft with the specified requirements is designed. The second study is to design a transport with the same requirements as an existing aircraft, and the two aircraft are compared. In order to evaluate the design requirements, trade-off studies are performed in the third case study.

The design work including input and output of data is done on a user interface with some windows as shown in Fig.6. The windows (a) and (b) display names of design steps and names of design tasks, respectively. Users can describe required design tasks in the form of message passing in the window (c). The output data are displayed in the window (d). The system displays various types of data as illustrated in Fig.7.



```
Sizing) ***
   'Value'->0.02

*** setSpecificFuelConsumption(Initial
Sizing) ***
   'Value'->0.5

*** passengerWeight(Initial Sizing) ***
   'Value'->600

*** estimateInitialTakeoffWeight(Initial
Sizing) ***
   'Value'->1000

*** initialSizing(Initial Sizing) ***
   'Value'->3307.2

*** wingLayout(Layout Data) ***
      Reference Wing Area(ft2) =360.8
      Wing Span(ft) =46.5247
      Root Chord(ft) =11.0743
      Tip Chord(ft) = 4.4313
      MAC(ft) = 8.22956
   an AircraftDesignTaskObject

*** fuselageLayout(Layout Data) ***
   'Value'->22.5706

*** verticalTailWingLayout(Layout Data)
***
```

```
span = designStepObject designTaskObject
          selectDesignVariable: 'Wing Span'.
cRoot = designStepObject designTaskObject
          selectDesignVariable: 'Root Chord'.
cTip = designStepObject designTaskObject
          selectDesignVariable: 'Tip Chord'.
aR = designStepObject designTaskObject
          selectDesignVariable: 'Aspect Ratio'.
taper = designStepObject designTaskObject
          selectDesignVariable: 'Taper Ratio'.
mac = designStepObject designTaskObject
          selectDesignVariable: 'Mean Aerodynamic Chord'.
area value:( w0 value / sW value ).
span value:( aR value * area value) sqrt.
cRoot value:( 2 * area value / span value /
   ( 1 + taper value)).
cTip value:( taper value * cRoot value).
mac value:( 2 / 3 * cRoot value * ( 1 + taper value
   + taper value squared) / ( 1 + taper value)).
```

Fig. 6 User interface

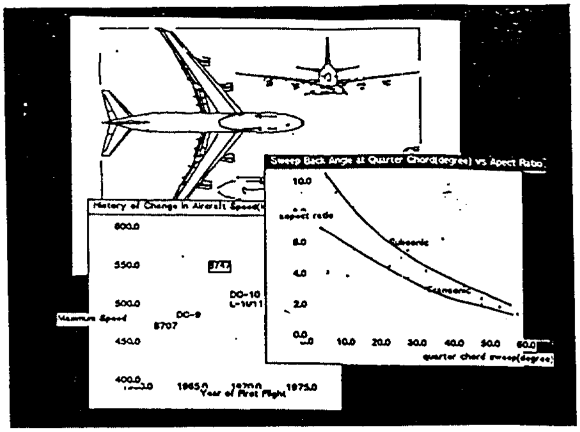


Fig. 7 Display of data

5.1 Case 1

The design requirements are shown in Table 3 and the design results are presented in Table 4. Design parameters are taken from the existing data for the same type of aircraft, so that the results may be acceptable.

Table 3 Requirements of Case 1

Items		Values
Range	(nm)	280
Number of Crew		1
Payload	(lb)	100
Takeoff Length	(ft)	1000
Maximum Speed	(kts)	130
Cruise Speed	(kts)	115
Cruise Altitude	(ft)	8000
Stall Speed	(kts)	50

Table 4 Design results of Case 1

Items		Values
Takeoff weight	(lb)	1390
Max horse power	(bhp)	160
Wing area	(ft)	150
Wing span	(ft)	30
Fuselage length	(ft)	18.5
Fuselage diameter	(ft)	3
Range	(nm)	285
Takeoff length	(ft)	900
Maximum speed	(kts)	120

5.2 Case 2

In this study, a design is carried out for the same specifications as YS-11 developed in Japan. Table 5 shows the data of YS-11 and the design results. The performed design satisfies the maximum speed and range requirements, but the size is bigger than that of the original aircraft. The reasons may be attributed to the error in the assumption of some design parameters.

Table 5 Design results of Case 2

		YS-11	Design results
Takeoff weight	(lb)	54000	59800
Max horse power	(bhp)	6300	6500
Range	(nm)	700	750
Maximum speed	(kts)	260	270
Takeoff length	(ft)	3000	2800
Wing area	(ft)	1020	1100
Wing span	(ft)	104	115
Fuselage length	(ft)	59	65

5.3 Case 3

Two trade-off studies on range versus take-off weight are made, and the results are given in Fig.8. Range has significant effect on take-off weight, and it must be set carefully. The system allows users to select design variables for trade-off studies if they wish.

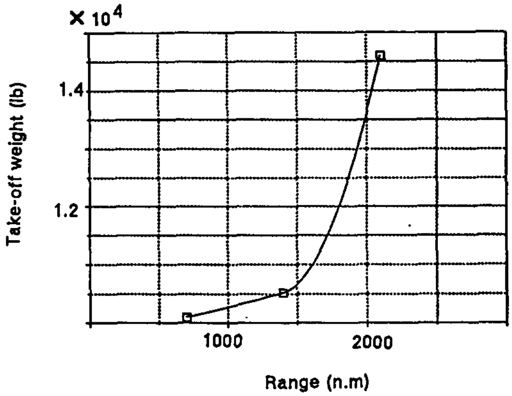


Fig. 8 Trade-off studies

6. Concluding remarks

The CAI system for the conceptual design of aircraft has been developed. Various kinds of the processing data and vagueness of the design process have serious influence on programming. Most of the modules based on the empirical data may have lack of accuracy. Nevertheless, they can be used for a conceptual design with the trial and error method. Extension of the system is easily accomplished by replacement and refinement of the modules. The results of the present study are summarized as follows:

(1) The main aspects of the conceptual aircraft design have been clarified, and the concepts of the CAI system are proposed and implemented.

(2) The CAI system for the conceptual aircraft design has been developed by using an Object-Oriented approach, which results in readable and flexible programs.

(3) The system manages various types of data, such as numerical values, symbols, graphs and drawings.

References

[1] J.Roskam, "What is Needed to Teach Aeronautical Engineering Students How to Design Aircraft?", AIAA 90-3257, AIAA/AMS/ASEE, Aircraft Design, System and Operations Conference, Dayton, OH., USA, Sept.17-19, 1990.

[2] Y.Murotsu and C.S.Park,"An Expert System for Instruction of Preliminary Aircraft Design", AIAA90-3261, AIAA/AMS/ASEE, Aircraft Design, System and Operations Conference, Dayton, OH., USA, Sept.17-19, 1990.

[3] Y.Murotsu, C.S.Park and M.Ichikawa, "A Development of Computer Support System for Wing Design", China-Japan Joint Symposium on Aerodynamics and Aircraft Design, Nanjing, CHINA, Sept.27-29, 1990.

[4] Y.Murotsu, S.Tsujio and C.S.Park, "A Development of Hypermedia Type Database System for Instruction of Aircraft Conceptual Design", 29th Aircraft Symposium, Gifu, JAPAN, Oct.7-9, 1991(in Japanese).

[5] Daniel P.Raymer, Aircraft Design: A Conceptual Approach, AIAA Education Series, Washington, D.C.,1989.

[6] A.Golberg and D.Robinson, Smalltalk-80, The Language and its Implementation, Addison-Wesley, 1983.

[7] Smalltalk-80 Virtual Image Version VI2.5, ParcPlace Systems Inc, 1987.