National Aerospace and Electronics Conference (NAECON), 20-24 May 1985, Dayton, Ohio

# An Expert System for Integrated Aircraft/Engine Controls Design

James H. Taylor

GE Corporate Research and Development
Control Technology Branch
Schenectady, New York 12345

Abstract - In this paper, we discuss the concepts, requirements and architecture for an expert system for integrated aircraft/engine control systems analysis and design. The purpose of this concept is to provide a high-level environment embodying expertise from the many fields that enter into integrated flight and engine controls design (aerodynamics, structures, propulsion, pilot/aircraft interaction, mission performance requirements, control system design and validation), thereby facilitating the design of integrated aircraft control systems and ensuring that the best possible designs are obtained. The expert system concept also has capabilities to handle data-base management, and to take maximum advantage of existing conventional software. A prototype expert system has been implemented that demonstrates many of the capabilities and benefits of such an environment. A record of a hypothetical controls system modeling, analysis, and design session is included in this paper.

## 1. INTRODUCTION

Expert or knowledge-based systems are software environments designed to aid in solving problems that require high levels of expertise, some degree of inference ("reasoning"), and the use of heuristics (nonrigorous procedures or "rules of thumb"). Such problems are generally complicated and broad in scope, and are not amenable to clear-cut well-posed algorithmic solutions. Control system engineering, in the broad sense considered here, is such a task. The types of expertise that are required for computer-aided control engineering (CACE) problem solving are: development and diagnosis of plant models, modifying the plant to render it more suitable for control if necessary (e.g., adding sensors and/or actuators), formulating a realistic design problem, selecting appropriate analysis and design methods, performing tradeoffs, validating and documenting the design, implementing the controller, and using conventional CACE software. Heuristics are a major factor in a human expert's ability to formulate a well-posed design problem, and reasoning capability is advantageous for directing and keeping track of the design process as it progresses. Also, expert systems provide a high-level, flexible, and supportive environment that can relieve the user of much of the low-level detail and drudgery involved in using a number of large software packages. For more information on expert systems, or for additional background reading in the area, refer to [1-2] or to [3] which provides a brief overview and a more extensive bibliography than that given here.

The development of software systems to capture the expertise of experienced human experts is currently attracting a great deal of attention [1-3]. The main reasons for this interest are that tasks being performed by humans are steadily becoming more complicated, that it is becoming increasingly clear that experienced human experts are in short supply, and that hardware and software technology have recently caught up with the requirements for developing meaningful expert systems. In many fields of engineering, the first factor is evident in the recent trend towards creating integrated software environments for complicated analysis and design procedures that require substantial experience for mastery. Again, one such area is CACE [4-9].

A discussion of expert systems technology leads to the idea of providing support for "less-than-expert users"; this requires careful consideration. First, there are several factors involved in this phrase: Such a user may not be aware of the latest theoretical developments in all fields involved in the problem to be solved, may not have had the experience required to synthesize theory into an effective analysis and design approach, and/or may not be a frequent practitioner of system design or user of the required software. Considering the breadth of knowledge required for integrated flight and engine control system analysis and design, it is probably not reasonable to expect that many users of a major software environment for this activity will be experts in all aspects of the problem being solved. Providing support for "less-than-expert users" in this sense is desirable, reasonable, and possible. On the other hand, it should be clearly understood that it is not appropriate to speak of developing a CACE environment for use by personnel with no knowledge of the field. With this understanding, it is fair to say that many less-than-expert users find that currently-available CACE software environments are difficult to use effectively, that the situation is worsening as such environments become more comprehensive, and that it is worthwhile correcting this situation.

In this paper, we specifically discuss the architecture and requirements for an expert system for integrated aircraft/engine control systems analysis and design. The purpose of this concept is to provide a high-level environment embodying expertise from the many fields that enter into integrated flight and engine controls design, i.e., aerodynamics, structures, propulsion, pilot/aircraft interaction, mission performance requirements, control system design and validation, thereby facilitating the design of integrated aircraft control systems and ensuring that the best possible designs are obtained. Such a system should aid the controls engineer in every aspect of aircraft control system design, from model development through control system design, validation, and implementation, and should permit determining the impact of alternative control system designs on aircraft mission performance and carrying out tradeoff analyses. This concept is based on the motivational issues, general functional requirements, and basic expert system ideas and architecture presented in [10,11]. The expert system CACE-III that embodies these ideas is, in our opinion, the first true <u>third-generation</u> CACE environment; hence its name.

The expert system architecture in [10] has also been conceived to handle data-base management, and to take maximum advantage of existing conventional software. In the area of integrated aircraft control systems analysis and design, data-base management includes keeping track of all the files containing models, analysis results, diagnostics, performance results, et cetera, along with the necessary relational information. Conventional software encompasses computer programs for aerodynamic modeling, structural modeling, engine modeling, nonlinear simulation, linearization (standard and quasi-linear) and model reduction, control systems design, validation, and implementation. This architecture is highly modular, to allow the introduction of new packages that may extend the expert system's capabilities or improve its performance (through the use of numerically superior algorithms, for example).

A generic prototype expert system has been implemented that demonstrates many of the capabilities and benefits of such an environment. A record of a hypothetical controls system modeling, analysis, and design session will be included, to make the concept more tangible and credible. With this background and evidence, we believe that the promise of this approach to provide a high-level design environment that is powerful, supportive, flexible, broad in scope, and readily accessible to non-expert users will be established.

In the remainder of this paper, we will overview an architecture for a third-generation CACE environment that is currently under development, outline the requirements for such a system, and illustrate its "performance" with an extended example (sample transaction). We will then close with a status report, summary, and conclusions.

## 2. AN EXPERT SYSTEM ARCHITECTURE FOR CACE

We have developed the architecture of an expert system for CACE by "modeling" the activity of a human expert, and will present it from the same vantage point. For a more detailed discussion, refer to [11].

One central issue in the control system design process is the <u>complete problem formulation</u>. This may be represented by a "list of facts" or, in artificial intelligence (AI) terminology, "frame". The information in this frame may be organized or partitioned into three components:

1. <u>Plant model characterization</u> - ordinary or partial differential equation models, linear or nonlinear, stable or unstable, minimum or nonminimum phase, (un)controllable and/or (un)observable, et cetera.
2. <u>Constraints</u> - architectural (e.g., centralized or distributed control), implementation (e.g., analog or digital), parametric constraints (e.g., gains, data rate limits), et cetera.
3. <u>Specifications</u> - time response, frequency response, performance indices, et cetera; sensitivity, disturbance rejection, robustness, et cetera.

This <u>problem frame</u> is a major focal point for CACE. A major distinguishing attribute of a human expert is the ease with which these data are assembled to obtain a meaningful, workable problem. Therefore, formulating this list of facts to obtain a well-posed problem must be a central concern in CACE-III. This list of facts is established via the interaction of the user and the expert system, as illustrated in Section 4, and may involve using a great deal of conventional software and keeping track of the resulting data base (e.g., model files, data files, records of the analysis and design methods used). We will provide an example of the problem frame list of facts at the end of the transaction presented below, where its structure and meaning will be more transparent.

In the second part of the execution of a control design problem, we believe that the human expert works in a parallel construct, which we call the <u>solution frame</u>. In CACE-III, this is a list of facts that is developed as a "scratch pad" where the expert system keeps track of what has been done and what needs to be done, information required for decisions about the selection of design procedures and tradeoff analysis (also data-base information), and a log of the entire transaction. Such a frame is not as clearly defined as the problem frame, because it is not as generic as the problem frame and because a set of automated design procedures has not yet been established.

The <u>facts</u> involved in the problem and solution frames form one key constituent of CACE-III. The other requirement is a rule base that mechanizes the CACE functions outlined in Section 1. A complete expert system for CACE for integrated flight and engine control must be able to carry out these tasks, and must know how to use conventional CACE software packages in so doing.

The line of thought provided by the above linking of the activities of an expert with two key lists of facts (frames) and associated rule bases gave rise to a complete functional structure of CACE-III that is depicted in Fig. 1. In particular, we have created a construct in which the rule base is partitioned into six parts, as shown. The functions of the rule bases may be summarized as follows:

1. RB1 governs interactions among the design engineer, plant models (component-level and integrated, partial or ordinary differential equation, nonlinear and linear), and the model component of the problem frame. This rule base provides support in model development (including diagnostics relating to the physical process and suitability of models for control system design and numerical analysis), and sees to it that all required plant data are added to the knowledge base.

2. RB2 governs interactions between the design engineer and the constraint and specification components of the problem frame. Constraints are requested but are not mandatory; if supplied, these are also written into the list of facts that makes up the problem frame. These rules guide the user in entering design specifications and checks specifications for consistency, completeness, and achievability (realism).

3. RB3 and RB5 govern interactions between the problem frame and the solution frame. RB3 deals with specifications, constraints, and plant characteristics, and initializes the list of facts in the solution frame describing what needs to be done to achieve design goals. If design iteration or tradeoff analysis has been called for, RB3 resets the solution frame accordingly. RB5 handles situations where specifications cannot be met or tradeoffs need to be considered, by supporting the user in modifying the problem frame appropriately.

4. RB4 governs interactions between the solution frame and the available design procedures. These rules decide what design approach(es) will best solve the problem, executes the appropriate procedure(s) and algorithm(s), and updates the solution frame to reflect the corresponding addition/change in the system. RB4 also performs a preliminary validation by checking that all specifications are met.

5. RB6 governs the final control system validation process (which generally involves highly realistic simulation or emulation of the plant and controller), conversion from idealized controller design to practical implementation, and documentation. The latter involves archiving a record of the design process, including tradeoffs and information supporting all design decisions, and a record of the data base (model and data files, including information regarding assumptions and conditions for validity).

The goal of the first two rule bases is to have a well-formulated problem, thus ensuring a reasonable probability of success in the design phase. RB3 serves to initialize the solution frame or reset it in the case of tradeoff or iterative design. Observe that RB3, RB4 and RB5 represent an iterative or dynamic "loop"; we pass through these rule bases until all specifications are satisfied, if possible. Finally, RB6 provides the final assurance that the user has a control system that will perform as required, with as little need as possible for additional engineering for implementation, and with the engineering documentation required for future configuration control. A more detailed discussion of the functions of each rule base is provided in [11]; much of this will be illustrated in the sample transaction provided in Section 4.

Note that these rule bases invoke conventional CACE software in performing all analysis and design functions. Expertise regarding the use of this software is "built-in", so that the user need not know command sets and syntaxes. However, it is important that the user be able to run this software manually, if desired, so that more experienced users can expedite the design process or "steer" it according to their experience or preferences. This need can be met quite readily, as described in [10,11], essentially permitting the user to operate CACE-III in "automatic", "semi-automatic" and "manual" modes, as desired.

## 3. REQUIREMENTS FOR CACE-III

The development of CACE-III requires expert system software and conventional software for simulation, analysis, and design. The latter should be selected primarily on the basis of functional requirements and quality, so that the software can perform the required operations with robust numerics; the user interface is of secondary importance, since the expert system will be managing that aspect.

The expert system software requirements have been discussed more fully in [12]. In summary, we are currently creating our demonstration or prototype version of CACE-III using DELPHI, a GE-developed LISP-based inference engine which is the successor to DELTA which we used in the beginning of our development effort. The latter inference engine was developed at GE for a diesel-electric locomotive trouble-shooting aid [13]. We have outlined the operation of DELTA to the extent required for tutorial purposes in [10,11]; DELPHI is similar in its handling of production rules. However, the DELPHI system has several additional or extended capabilities that are essential to our application:

a. the ability to run and exchange information with external processes (conventional software),

b. the ability to assign symbolic or numerical values to logical variables, and

c. the ability to perform arithmetic operations.

Additional capabilities that are very useful are: high-level rule-base debugging tools [11], an effective 'why' facility, and informative displays. We have not pursued any of the latter items in any detail at this point in our effort.

## 4. A SAMPLE CACE-III TRANSACTION

An effective way to manifest the capabilities and behavior of a system such as CACE-III is to present a sample transaction or session log. Based on the rule base development to this point in the project, we can hypothesize quite detailed dialogs. Much of this transaction can actually take place today, in at least a rudimentary fashion; however, portions of it (especially those specifically related to integrated flight and engine control) are not possible. We have included as many concepts as we can, to illustrate what we have done and what we have in mind. Perhaps some of this transaction will strike the reader as being simplistic - however, the primary intent of this illustration is to clarify ideas. Also, most of the dialog (as in previous discussions [10]) is in classical controls terminology, to make the illustration accessible to the largest possible audience. We are not espousing any particular design approach or software by the choice of procedures used in this example.

A good demonstration problem for many of our ideas is provided by the task of designing a controller for a nonlinear model of a fighter aircraft. Note that the flight conditions and other technical information are "made-up" and not meant to be representative of any particular aircraft or flight regime.

NOTE: All CACE-III output/response is in normal type in this presentation, while user input is <u>underlined</u>. Comments are provided parenthetically.

```
Welcome to CACE-III

QUERY: Is this a new session? (Y or N) >>  No

CONTINUE CONTROL SYSTEM ANALYSIS AND DESIGN SESSION

Enter session file name (fname.MDB) >>  Fnnn.MDB

QUERY: Do you want a synopsis? (Y or N) >>  Yes

MASTER FILE Fnnn.DBF * 27 FEB 1985 * [TAYLOR.PROB7]

Aircraft: Fnnn Fighter
   Primary model file name: FnnnNLM.ODE
      (nonlinear ordinary differential eqn. model)
   Nonlinear model diagnosis data for
      FnnnNLM.ODE is in file FnnnNLM.DIA.
   Secondary linearized models:
      FnnnLM01.ABC  linearized at
         p  =  0.0,  0.0,  3. km
         v  = 12.2,  0.0,  0.0 m/s
         dθ =  0.0,  0.0,  0.0 deg/ss
         ....
         da = 5.1  deg, de = ....
      FnnnLM02.ABC  linearized at
         p  =  0.0,  0.0,  1. km
         v  = 10.3,  3.0,  3.7 m/s
         dθ = 6.7, 1.5,  3.5 deg/s
         ....
         da = 9.3  deg, de = ....

   Linear/nonlinear diagnosis of
      FnnnLM01.ABC is in file FnnnLM01.DIA.
   Controller design for FnnnLM01.ABD
      is in file FnnnLM01.CTR.
   Performance data for FnnnLM01.CTR
      is in file FnnnLM01.PER

   Linear/nonlinear diagnosis of
      FnnnLM02.ABC is not done.
   Controller design for FnnnLM02.ABC
      is not done.

Engine:  GE-F110
   Primary model file name: GEF110NLM.ODE
      (nonlinear ODE model)
   Secondary linearized models: none
   Nonlinear diagnosis of GEF110NLM.ODE
      is not done.

Structure:  nil
   Primary model file name: (none)

END OF MASTER FILE Fnnn.DBF * * * * * * * * * * *

Menu for session continuation:

      1. Model development
      2. Model diagnosis
      3. Constraint development
      4. Specification development
      5. Control system design
      6. Tradeoff analysis
      7. Control design validation
      8. Terminate session.    SELECT:  2

Enter model name >> Fnnn02LM.ABC

BEGIN DIAGNOSIS OF MODEL FnnnLM02.ABC

   (The following activities are governed by RB1.)

   (The following section of this dialog represents
   a typical diagnosis of a linearized plant.  All
   required analysis software exists.)

The operating point is: p  =  0.0,  0.0,  1. km
                        v  = 10.3,  3.0,  3.7 m/s
                        dθ = 6.7, 1.5,  3.5 deg/s
                        ....
```

```
                                 da = 9.3  deg, de = 4.5 deg,
                                 dr = 6.0  deg
```

(CACE-III terminology below is based on standard notation: u = input, x = state, y = output.)

```
Enter actuator ranges, + du >>  3.0 1.5 0.0
```

(This information is needed in order to diagnose the importance of nonlinearity.)

(CACE-III finds the eigenvalues of that model and prepares data for entry in the problem frame, writing to file FnnnLM02.DIA, and a report to the user. Based on the eigenvalues, CACE-III chooses $\omega_0$ and tsim and simulates both linear and nonlinear models for $0 < t < \text{tsim}$ with $u_1 = 9.3 + 3 \sin \omega_0 t$, $u_2 = 4.3 + 1.5 \sin \omega_0 t$, $u_3 = 6.0$, where the input amplitudes are given by the operating point and du requested above and $\omega_0$ is based on the linearized plant eigenvalues. CACE-III then obtains the Fourier coefficients and distortion information needed to prepare a report on the degree of nonlinearity. Based on these analyses, the following report is made:)

```
DIAGNOSIS OF LINEAR MODEL Fnnn02LM.ABC:

Quantitative results:

   Fourier analysis results for input du * sin (2.4t):
      maximum percent second harmonic: 0.2
      maximum percent third harmonic: 3.21
   Linearized model eigenvalues at u0, x0:
      s1 = -0.554 + 1.154j    Short period
      s2 = -0.554 - 1.154j
      s3 = -0.145 + 1.684j    Dutch roll
      s4 = -0.145 - 1.684j
      s5 = -0.588             Roll
      s6 = 0.0004 + 0.132j    Phugoid
      s7 = 0.0004 - 0.132j
      s8 = -0.069             Spiral
      s9 = 0.0
   Linearized model distortion analysis results
   for step inputs of amplitude du:
      maximum percent rms distortion: 1.36

Qualitative results:

The FnnnNLM.ODE system model has no discontin-
   uities for the conditions specified.
The FnnnNLM.ODE system model is only mildly
   nonlinear over the input range specified.
The Fnnn02LM.ABC linear model is adequately
   accurate for the conditions specified.
The Fnnn02LM.ABC model has 2 unstable modes.
The Fnnn02LM.ABC model is controllable.
The Fnnn02LM.ABC model is observable.

END OF DIAGNOSIS.

The linear/nonlinear model diagnosis results have
been written to file Fnnn02LM.DIA.

QUERY: Do you wish more diagnosis (Y or N) >>  No

Menu for session continuation:

         1. Model development
         2. Model diagnosis
         3. Constraint development
         4. Specification development
         5. Control system design
         6. Tradeoff analysis
         7. Control design validation
         8. Terminate session.    SELECT:  4

BEGIN SPECIFICATION DEVELOPMENT FOR FnnnLM02.ABC
```

(At this point RB2 takes control.)

```
WARNING: Without constraints the design you realize
         may be idealistic.
```

(The following dialog represents existing specification entry procedures.)

```
MENUS TO ENTER SPECIFICATIONS

Menu for sensor type

         1. Continuous-time.
         2. Discrete-time.
         3. Mixed
         4. Terminate session.    SELECT:  1

Menu for controller type:

         1. Continuous-time.
         2. Discrete-time.
         3. Mixed
         4. Return to preceding menu.    SELECT:  1

Menu for control system structure

         1. Single-input single-output
         2. Decoupled (diagonally dominant).
         3. Decentralized controller
         4. Hierarchical
         5. Unconstrained
         6. Return to preceding menu.    SELECT:  5

SPECIFICATION DEVELOPMENT FOR UNCONSTRAINED CONFIG

Menu for type of continuous-time spec

         1. Locations of poles in s-plane.
         2. Time response.
         3. Frequency response.
         4. No more specs to enter.
         5. Return to preceding menu.    SELECT:  1

Menu for type of s-plane spec

         1. Maximum real part of s.
         2. Minimum damping ratio.
         3. Value of dominant pole (Re,Im).
         4. Damping ratio and wn of dominant poles.
         5. Return to preceding menu.    SELECT:  2

Enter min damping ratio of closed-loop poles >> 0.8

WARNING: Your spec for min damping of closed-loop
         poles is not reasonable.  It should not be
```

```
        less than 0.59 .

QUERY: Do you wish to modify? (Y or N) >>  Why

   (The user can enter "Why" any time to obtain an
   explanation of CACE-III statements or actions.)

   A pole-placement algorithm was applied to the
   linearized model of the plant; it was unable to
   place the poles with damping greater than 0.59
   without excessive gains based on the plant input
   operating range you specified.  RULE RB2-1022.

QUERY: Do you wish to modify? (Y or N) >>  Yes

Enter min damping ratio of closed loop poles >> 0.59

   Your specification is reasonable.

Menu for type of continuous-time spec

        1. Locations of poles in s-plane.
        2. Time response.
        3. Frequency response.
        4. Structural, e.g., no. of leads.
        5. No more specs to enter.
        6. Return to preceding menu.    SELECT:  3

   (Specification development proceeds until a
   valid set of specifications is obtained; we
   skip over similar interactions.  See [10] for
   a more complete presentation of this activity.)

   You have now entered specifications that appear
   to be complete, consistent, and realistic.

QUERY: Do you wish to modify specs? (Y or N) >>  No

Enter specification tolerance (percent) >>  10.0


Menu for session continuation:

        1. Model development
        2. Model diagnosis
        3. Constraint development
        4. Specification development
        5. Control system design
        6. Tradeoff analysis
        7. Control design validation
        8. Terminate session.    SELECT:  5

BEGIN CONTROL SYSTEM DESIGN FOR MODEL FnnnLM02.ABC

   (RB3 initializes the solution frame list of
   facts; CACE-III control passes to RB5.)

WARNING: Without constraints the design you realize
         may be idealistic.

   (The following represents an actual design
   that can be performed by CACE-III [13].)

   Your specifications suggest using the classical
   frequency-domain approach (CFDA) to design a
   lead-lag compensator for this loop.

QUERY: Do you wish to use CFDA ? (Y or N) >>  Yes

   (There is a detailed report made to the user
   as control system design proceeds, displaying
   compensator blocks and frequency response plots
   as the forward path is modified - see [14] which
   describes our controller design algorithm.)

CFDA CONTROLLER DESIGN IS COMPLETE.

Specifications have been met.  CFDA achieved:

closed loop poles:

         s1  = -2.64 + 2.314j
         s2  = -2.64 - 2.314j
         s3  = -1.792
         s4  = -1.75 + 2.390j
         s5  = -1.75 - 2.390j
         s6  = -1.383
         s7  = -1.29 + 1.555j
         s8  = -1.29 - 1.555j
         s9  = -0.614
         s10 = -0.151
         s11 = -0.052

QUERY: Do you wish to modify? (Y or N) >>  No

END OF CONTROL SYSTEM DESIGN


Menu for session continuation:

        1. Model development
        2. Model diagnosis
        3. Constraint development
        4. Specification development
        5. Control system design
        6. Tradeoff analysis
        7. Control design validation
        8. Terminate session.    SELECT:  7

BEGIN CONTROL DESIGN VALIDATION FOR FnnnLM02.ABC

   (CACE-III control passes to RB6.)

WARNING: Without constraints you cannot obtain a
         meaningful validation.

Menu for control system validation:

        1. Define control system implementation
        2. Enter/modify constraints
        3. Enter/modify specifications
        4. Perform control simul with FnnnLM01.ABC
        5. Perform control simul with FnnnNLM.ODE
        6. Perform controller emul with FnnnLM01.ABC
        7. Perform controller emul with FnnnNML.ODE
        8. Terminate session.    SELECT:  1

QUERY: Is it analog or digital (A or D) >>  D


WARNING: Without constraints you cannot obtain a
         meaningful validation.

Menu for control system validation:

        1. Enter/modify constraints
        2. Enter/modify specifications
```

3. Define control system implementation
4. Perform control simul with FnnnLM01.ABC
5. Perform control simul with FnnnNLM.ODE
6. Perform controller emul with FnnnLM01.ABC
7. Perform controller emul with FnnnNML.ODE
8. Terminate session. SELECT: 1

(User defines constraints, returns to validation, and selects item 5, nonlinear simulation.)

Menu for simulation condition definition:

1. Step resp with full actuator range (du)
2. Step resp with half actuator range (.5du)
3. Sine resp with full actuator range (du)
4. Sine resp with half actuator range (.5du)
5. Other (next menu)
6. Terminate session. SELECT: 1

NONLINEAR SIMULATION COMPLETED.

QUERY: Do you wish resp plots? (Y or N) >> Yes

(We omit these for the sake of space. The validation procedure we are implementing includes simulation of the continuous-time controller with the linearized plant model, then with the nonlinear model; then discretization of the controller and simulation of the digital controller with the nonlinear plant. This last simulation would proceed from a simple discrete-time model of the controller to a rigorous implementation that took word length and computation time into account.)

QUERY: Do wish a μproc implementn? (Y or N) >> Y

Enter microprocessor type >> 8086

IMPLEMENTATION COMPLETE

QUERY: Do wish an emulation? (Y or N) >> Yes

QUERY: Do wish to download code? (Y or N) >> Yes

(Based on the current state-of-the-art, the last three steps may be fanciful in most circumstances, but this is a possible - and highly desirable - outcome that completes the "control system design problem" suggested in Section 1.)

The status of a session, or the outcome of a session when completed, is embodied in the list of facts that has been written in the course of the transaction. Such a list is illustrated in Table 1, which depicts the problem frame or list of facts that might exist in the CACE-III knowledge base at the end of a MODEL, DIAGNOSE and SPECIFY session. These facts are an intermediate outcome of the transaction presented above; a reading of the transaction will provide a clear explanation and interpretation of this list of facts, so we will not comment on them further here. It should be observed that the underlying *data*, e.g., the numerical results of the nonlinear system diagnosis and the linearized model and its diagnosis, are contained in ancillary files. The expert system does not use this type of data directly, but it must know where such information can be found so that it can be provided to external analysis and design procedures and used for data-base management as required.

Table 1

THE PROBLEM FRAME AFTER MODELING, DIAGNOSIS, AND SPECIFICATION DEVELOPMENT

| | FACT | | BASIS |
|---|---|---|---|
| AIRCFT-NL-MDL | FNAME | FnnnNLM.ODE | User |
| AIRCRAFT | MODEL | NONLINEAR | Inferred |
| ENGINE-NL-MDL | FNAME | GEF110NLM.ODE | User |
| ENGINE | MODEL | NONLINEAR | Inferred |
| AIRCFT-NL-MDL | TIME-TYPE | CONT | Inferred |
| AIRCFT-NL-MDL | STATE-TYPE | CONT | Inferred |
| AIRCFT-NL-MDL | ORDER | 9 | Inferred |
| AIRCFT-NL-MDL | INPUTS | 3 | Inferred |
| AIRCFT-NL-MDL | OUTPUTS | 3 | Inferred |
| AIRCFT-NL-MDL | DIAGNOSIS | DONE | Inferred |
| AIRCFT-NL-MDL | DIA-FNAME | FnnnNLM.DIA | Inferred |
| AIRCFT-NL-MDL | NL-BEHAVIOR | MILD | Inferred |
| AIRCFT-L-MDL1 | FNAME | FnnnLM01.ABC | Inferred |
| AIRCFT-L-MDL2 | FNAME | FnnnLM02.ABC | Inferred |
| AIRCFT-L-MDL1 | STABLE | NO | Inferred |
| AIRCFT-L-MDL1 | CONTROLLABLE | YES | Inferred |
| AIRCFT-L-MDL1 | OBSERVABLE | YES | Inferred |
| AIRCFT-L-MDL1 | DIAGNOSIS | DONE | Inferred |
| AIRCFT-L-MDL1 | DIA-FNAME | FnnnLM01.DIA | Inferred |
| AIRCFT-L-MDL2 | DIAGNOSIS | DONE | Inferred |
| AIRCFT-L-MDL2 | DIA-FNAME | FnnnLM02.DIA | Inferred |
| SENSOR | TIME-TYPE | CONTINUOUS | User |
| CONTROLLER | TIME-TYPE | DISCRETE | User |
| CONTROLLER | STRUCT | UNCONSTRAINED | User |
| MIN-DAMPING | LIMIT | 0.59 | Inferred |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| SPEC-SESSION | TERMINATION | NORMAL | Inferred |

## 5. STATUS

Thus far we have developed the following:

1. an architecture for CACE-III, including a specific, detailed outline of the functional characteristics of the expert system and of the rule base (Fig. 1; [10-12]);

2. several detailed "transactions" or dialogs between CACE-III and a user, to provide a more tangible "bottom-up" basis for detailing the functions of the various rule bases shown in Fig. 1; and

3. working rule bases for several functions, including nonlinear system model diagnosis, equilibrium finding and linearization, linear system model diagnosis, specification development, an automatic control system design procedure (lead-lag compensator design for a single-input/single-output plant), and

some aspects of validation (simulation with linear and nonlinear plant models).

## 6. SUMMARY AND CONCLUSIONS

### 6.1 Summary

The project we have described is still in the concept development phase. While we have a working expert system that can perform major parts of the problem creditably (e.g., designing a controller using a conventional frequency-domain design package; this activity alone requires 70 rules which cause about 100 commands to be issued to the design package in the course of one design [14]), there is clearly a great deal of detail that needs to be filled in and implemented before we have a complete prototype "real system". In the course of implementation, it is possible that the concepts we have outlined will continue to be refined and evolve as we understand the problem more completely.

Despite the incompleteness of this effort, there are several areas in which we believe significant progress have been made:

1. showing how expert systems may be used to provide meaningful solutions to the problems identified in Section 1;
2. developing a detailed architecture (Fig. 1), based on lists of facts (problem and solution frames), rule bases, and conventional CACE analysis and design software, that can support the user in the core activities of CACE outlined in Section 1; and
3. implementing enough of the rule base to be able to provide a very detailed picture of the capabilities of CACE-III (Section 4).

### 6.2 Conclusions

Most existing broad-scope CACE software environments are difficult for the less-than-expert and/or less-than-everyday user to master. Critical areas of user support required by non-expert users are the following: developing and diagnosing useful plant models, developing meaningful specifications, selecting design approach(es), performing tradeoff studies, validating the design, and implementing it. These services are not provided by existing CACE software environments to the extent required; we believe that the expert systems approach can fulfill these needs.

An expert system for CACE should be conceived with the following primary goals: expediting and removing as much drudgery as possible from the design process, reducing the probability of error, allowing the less-than-expert user to obtain better designs, adding better discipline and documentation to the design process, and enhancing rather than replacing engineering skill and judgment. It must be understood that no foreseeable expert system is going to provide the optimal solution to every control design problem, and that unqualified personnel will not be converted into experts.

In short, expert systems show great potential for providing the basis for a vastly improved man/machine environment for CACE, both generically and in the integrated flight and engine control arena where the multi-disciplinary aspect of problem solving is so important a consideration.

Acknowledgements

The work outlined here owes a great deal to the suggestions and efforts of a number of people aside from the author. Prof. Dean Frederick and Lt.-Col. John James, co-authors of [10,11,12,14], have played a major role in this research. Artificial intelligence expertise and expert system software provided by Drs. Piero Bonissone and Mel Simmons of the General Electric Corporate R and D Knowledge Based Systems Branch have also been extremely valuable. Many discussions with Lt. Mark Schiller, Air Force Wright Aeronautical Laboratories Flight Dynamics Laboratory, contributed significantly to concepts pertaining to integrated flight control. The ideas presented in this report have profitted substantially from these and other interactions.

### References

1. F. Hayes-Roth, D. Waterman, and D. B. Lenat, Building Expert Systems, Addison Wesley, Reading, MA, 1983.
2. D. Michie, Introductory Readings in Expert Systems, Gordon and Breach, London, 1982.
3. W. B. Gevarter, "Expert Systems: Limited but Powerful", IEEE Spectrum, August, 1983.
4. Special Issue on Computer-Aided Design of Control Systems, IEEE Control Systems Magazine, 2, No. 4, December, 1982.
5. D. K. Frederick, "Computer Programs for the Simulation and Design of Control Systems", Arab School on Science and Technology, Bloudan, Syria, September, 1981. (To appear in Advances in Control Systems, Theory and Application, Edited by J. B. Cruz, Jr., JAI Press, Greenwich CT, September, 1984.)
6. Proceedings of the Second IFAC Symposium CAD of Multivariable Technological Systems, Purdue University, West Lafayette, IN, September, 1982.
7. CACSD '83 Abstracts, IEEE Control System Society Symposium on CACSD, Cambridge, MA, September, 1983.
8. J. H. Taylor, "Environment and Methods for Computer-Aided Control Systems Design for Nonlinear Plants", Proc. Second IFAC Symposium CAD of Multivariable Technological

Systems, Purdue University, West Lafayette, IN, September, 1982.

9. H. A. Spang, "The Federated Computer-Aided Control Design System", Proceedings of the IEEE, 72, December 1984.

10. J. H. Taylor, D. K. Frederick, and J. R. James, "An Expert System Scenario for Computer-Aided Control Engineering", Proc. American Control Conference, San Diego, CA, 1984, pp. 120-128.

11. J. H. Taylor and D. K. Frederick, "An Expert System Architecture for Computer-Aided Control Engineering", Proceedings of the IEEE, 72, December 1984.

12. J. R. James, J. H. Taylor and D. K. Frederick, "An Expert System Architecture for Coping With Complexity in Computer-Aided Control Engineering", to appear in Proc. 3rd. IFAC Symposium on CAD in Control and Engineering Systems, Lyngby, Denmark, 1985.

13. H. E. Johnson and P. P. Bonissone, "Expert System for Diesel Electric Locomotive Repair", Journal of FORTH Applications and Research, 1, September, 1983.

14. J. R. James, D. K. Frederick, and J. H. Taylor, "On the Application of Expert Systems Programming Techniques to the Design of Lead-Lag Compensators", to appear in Proc. Control 85, Cambridge, England, 1985.

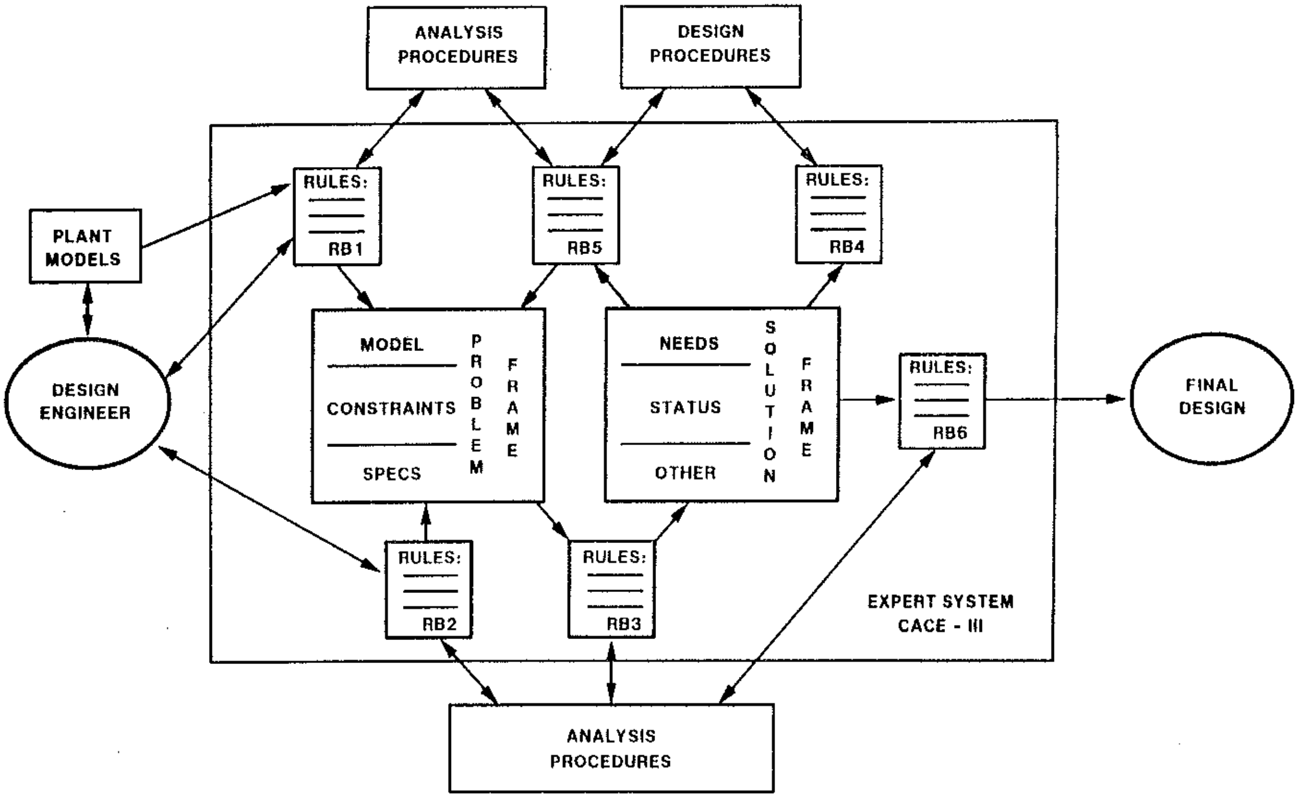

Figure 1. Complete Functional Structure of CACE-III