N89-19827

NESSUS/EXPERT: BRIDGING THE GAP BETWEEN ARTIFICIAL INTELLIGENCE AND FORTRAN

Pamela K. Fink, Ph.D.
Karol K. Palmer

Southwest Research Institute
6220 Culebra Rd.
San Antonio, Texas 78284
(512) 522-3268

ABSTRACT

The development of a probabilistic structural analysis methodology (PSAM) is underway at Southwest Research Institute (SwRI) as part of a research program for NASA/Lewis. In the near-term, the methodology will be applied to designing critical components of the next generation space shuttle main engine. In the long-term, PSAM will be applied very broadly, providing designers with a new technology for more effective design of structures whose character and performance are significantly affected by random variables.

The software under development to implement the ideas developed in PSAM resembles, in many ways, conventional deterministic structural analysis code. However, several additional capabilities regarding the probabilistic analysis makes the input data requirements and the resulting output even more complex. As a result, an intelligent front- and back-end to the code is being developed to assist the design engineer in providing the input data in a correct and appropriate manner. The type of knowledge that this entails is, in general, heuristically-based, allowing the fairly well-understood technology of production rules to apply with little difficulty. However, the PSAM code, called NESSUS, is written in FORTRAN-77 and runs on a DEC VAX. Thus, the associated expert system, called NESSUS/EXPERT, must run on a DEC VAX as well, and integrate effectively and efficiently with the existing FORTRAN code. This paper discusses the process undergone to select a suitable tool, identify an appropriate division between the functions that should be performed in FORTRAN and those that should be performed by production rules, and how integration of the conventional and AI technologies was achieved.

1 INTRODUCTION

1.2 Background To The Problem

Structural analysis techniques traditionally have been based on *deterministic methods*. That is, design characteristics such as geometry, material properties and loads were assumed to be constant. In reality however, there are factors such as manufacturing processes and operating conditions which introduce variance into design characteristics. In the past, this variance has either been ignored or a worst-case scenario has been adopted. Ignoring the variance may lead to design failure, the cost of which may be very great, perhaps involving the loss of human life. Adopting a worst-case scenario typically yields an extremely conservative and expensive design.

To account for the variability in design, structural analysis methods must step beyond the limitations of deterministic methods. The use of probabilistic structural analysis methods yields the ability for the designer to identify the relationships between specific design features and risk of structural failure. This provides information for educated decisions concerning risk, cost and need.

1.2 A Probabilistic Approach to Structural Analysis

A probabilistic structural analysis methodology (PSAM) is being developed at Southwest Research Institute (SwRI) as part of a research program for the National Aeronautics and Space Administration (NASA). In the near-term, the methodology will be applied to design critical components of the next generation space shuttle main engine. In the long-term, PSAM will be applied very broadly, providing designers with a new technology for more effective design of structures whose character and performance are significantly affected by random variables.

The objective of PSAM is to establish a computer-based methodology for design modeling of the real world effects of variability in applied loading (pressure, temperature, centrifugal force, etc.), material characteristics, tolerances, fits (boundary conditions), and strength data. The resulting computer programs will enable the design engineer to model how these design input variations (random variables) lead to variations in structural performance - stress, buckling, load, vibration amplitude, etc. The risk of structural failure can then be calculated by comparing the variation in performance with the variation in the allowable design limit condition, based on experience and material properties.

One goal of this NASA effort is to extend the research on probabilistic structural analysis methods and to develop a set of FORTRAN modules that embody the results of this research. The resulting system, called NESSUS (Numerical Evaluation of Stochastic Structures Under Stress) will include a module which performs standard deterministic structural analysis, a module which performs probabilistic analysis, a preprocessing module which will convert raw data to information that is usable by the probabilistic module, and a module which will perform a preliminary analysis of the results of the probabilistic module.

1.3 Where Artificial Intelligence Is Needed

The ultimate goal of the project is to go a step beyond building a set of analysis modules to creating an integrated, user-friendly structural design package. However, the NESSUS structural analysis modules alone constitute only a set of state-of-the-art structural analysis programs. To upgrade these modules to a design package they must be integrated into one system and a user interface must be added. Without the interface the design engineer must proceed through a time consuming and complex set of steps to use the NESSUS modules. The engineer must first build a data deck which contains a description of the finite element model of the structure to be analyzed. Though this data deck may consist of several thousand lines of data and must be arranged in a specific format for the NESSUS modules, the type of information it contains is generally required by any conventional structural analysis program. The next step, involving probabilistic analysis, however, will be new to most design engineers. These data requirements for NESSUS are very challenging, generally far exceeding that needed for the conventional, deterministic design approach. Once the deterministic and probabilistic portions of the data deck have been built, the engineer must execute the appropriate NESSUS module(s). The output from a NESSUS run consists of a file full of numbers which must be interpreted and analyzed. Finally, the engineer must detemine, from the results, what step to take next.

These are complex tasks which require expertise not only in finite element modeling, but also in the use of both the conventional and probabilistic NESSUS analysis modules. This expertise, however, is scarce since most engineers do not have experience with probabilistic analysis and even fewer have experience with the NESSUS modules. In addition to being complex, these tasks can be very time consuming. The analysis programs tend to run on large, number-crunching machines, such as the Cray, and can take many hours to complete a single run. Thus, a considerable amount of engineering time could be spent debugging erroneous data decks. These are classic reasons for using an expert system approach. An additional argument for using an expert system for this effort is that NESSUS is an evolving application, thus any interface to NESSUS must be flexible, expandible and maintainable. These are attributes which a rule-based expert system would provide. Therefore, it was decided to investigate the use of an expert system to serve as an aid to the design engineer in creating an appropriate and correct data deck and in analyzing the results of an analysis run. This expert system is called NESSUS/EXPERT.

2 IDENTIFYING THE REQUIREMENTS

2.1 Functional Requirements

NESSUS/EXPERT must serve as a flexible, user-friendly, integrated interface to the NESSUS structural analysis modules. In support of this function NESSUS/EXPERT must 1) be menu-driven and present to the user only those activities which are appropriate given the current status of the system, 2) invoke the various NESSUS modules directly at the appropriate times, 3) allow the user to leave the system at any stage with no loss of information, and 4) cater to users with varying degrees of expertise. In addition to these general system-wide requirements, the functional requirements of NESSUS/EXPERT can be categorized into two major functions: an intelligent user interface front-end and an intelligent back-end analysis aid.

2.1.1 An Intelligent Front-End to NESSUS

The front-end to the NESSUS structural analysis modules essentially provides an enhanced, on-line, automated user's manual. The NESSUS modules expect as input a data deck containing all of the structural information in a specific format. For large jobs this data deck may contain several thousand lines of data. To ease the chore of entering this data, NESSUS/EXPERT must infer data where possible and use defaults where available, informing the user of the values being used and their impact on the analysis.

To generate portions of the structural data required in the data deck, engineers often use independent software packages. For example, the nodes and elements of the finite element topology may be generated via a finite element preprocessor. To accommodate this practice, NESSUS/EXPERT must be able to read data in from existing files. Not all data will exist in files however, so NESSUS/EXPERT must also provide for manual entry of data. Regardless of the entry method used, NESSUS/EXPERT must allow the user to view and modify the data once it has been entered.

In serving as the front-end to the NESSUS structural analysis modules, NESSUS/EXPERT must provide guidance and advice to the user. This is probably the most important of NESSUS/EXPERT's functions given the size and complexity of this task. In order to best assist the user, NESSUS/EXPERT must

- prompt for all required information, being as specific as possible,
- activate and deactivate available options based on the current state of the job so that no incompatible selections are available to the user and so that all dependencies are represented,
- check for completeness of the data,
- check for inconsistencies between pieces of data,
- provide helpful hints involving idiosyncrasies of the NESSUS code and
- offer advice on optimal strategies in a given situation.

The final function required of NESSUS/EXPERT in its capacity as the front-end to the NESSUS structural analysis modules is to automatically generate the data deck needed by NESSUS from the data that has been entered. By providing the user with guidance and advice and preparing the data deck for the user, NESSUS/EXPERT will minimize the time spent by the engineer debugging the data deck.

2.1.2 An Intelligent Back-End Analysis Aid to NESSUS

The second major function of NESSUS/EXPERT is that it must serve as the back-end to the NESSUS structural analysis modules. The output from the structural analysis modules is simply a large file of numerical data. NESSUS/EXPERT must aid the user in analyzing and interpreting these results. Based on the results of the analysis, NESSUS/EXPERT must provide guidance to the user on what steps to take next. Recommendations might include modifying certain of the input data and rerunning the anlysis to see the effects, or observing that a particular parameter is sensitive to certain variations in the probabilistic data. NESSUS/EXPERT must also provide helpful hints involving idiosyncrasies of the NESSUS code and optimal strategies in given situations.

2.2 The Challenge: Integrating AI Into The FORTRAN Engineering Environment

The types of knowledge that must be embodied in NESSUS/EXPERT given the functional requirements identified include both factual knowledge and knowledge which involves rules-of-thumb and heuristics related to experience gained in using the NESSUS system. Thus, the knowledge that had to be captured fit, in a fairly straightforward manner, the production rule knowledge representation technique. The challenge, however, came in satisfying the operational requirements involved with functioning in an engineering environment and still staying within time and budget constraints. These operational requirements are that NESSUS/EXPERT must run on a VAX under VMS, be able to run interactively, real-time on large amounts of data, must be packaged for easy distribution, and be delivered in FORTRAN-77 to insure portability and availability to the engineering community.

3 SELECTING AN APPROACH

3.1 Potential Approaches To The Development Of NESSUS/EXPERT

Given only the functional requirements of the system, the selected approach to the problem would have been simple enough since most expert system building tools support this type of knowledge representation scheme. However, since very few expert system building tools are capable of extensive interaction with external functions, especially those written in FORTRAN, it was necessary to identify and investigate the approaches that might satisfy both the functional and operational requirements.

One potential approach was to use an existing expert system building tool written in FORTRAN. At the start of this project over 2 years ago, there were only two such tools. These lacked sufficient sophistication in rule structures and manipulation due to the difficulties involved in performing inherently recursively-based tasks, such as parsing and analysis, in a non-recursive language. An added difficulty was that there were licensing problems involved in using a commercial product to develop a system that would be distributed within the government for free.

Another approach considered was to write our own expert system building tool in FORTRAN. Consideration was given to writing an OPS-like production system in FORTRAN, but it was determined that the effort to do an adequate job was well beyond the scope of the project. Such an effort would leave little time and money for the development of the expert system itself.

The possibility of writing the expert system itself in FORTRAN was also rejected. FORTRAN does not provide the non-algorithmic constructs and pattern matching capabilities needed for efficient expert system development. The effort required to develop an expert system in a language that does not provide these capabilities would be tremendous. Further arguments against using FORTRAN as the development language for NESSUS/EXPERT were that it would probably be unacceptably inefficient and would loose many of the primary benefits of AI technology, such as flexibility, expandability, and maintainability.

The fourth option considered was to use an existing non-FORTRAN-based expert system building tool that could interface to FORTRAN and address the problem of delivery in FORTRAN later in the project. This temporary solution would allow for progress to be made in the development of the expert system thus enabling a better understanding of the effects the operational requirements were going to have on the system. A search was made to find an existing, inexpensive, fairly well-supported expert system building tool that could aid in the writing of production rules as well as readily interface to FORTRAN code.

No such system was found. Thus, the requirement that the tool be able to interface with FORTRAN code was reduced to being able to interface with the operating system and/or file system. Thus a public domain version of OPS5 was chosen for experimentation. This version was written in Franz LISP so the environment was not ideal. However, it did run on a VAX and it allowed rapid prototyping thus facilitating time effective experimentation with the interface between the expert system and the existing NESSUS code. The interface with FORTRAN was simulated using access to files. OPS5 proved to be sufficient in terms of the functional needs of the expert system itself but, as expected, not in the operational issues of integration and delivery.

Shortly after the development of the NESSUS/EXPERT prototype, a version of OPS5 written in BLISS was offered by DEC. This version provided the flexibility and power of the OPS5 language but not the problems involved with running in the Lisp environment. DEC OPS could interface reasonably well with the DEC FORTRAN on the VAX. The prototype of NESSUS/EXPERT was converted to DEC OPS to test the FORTRAN communication capabilities.

The resulting system was satisfactory from the standpoint of integration into the FORTRAN environment. However, inefficiency of the DEC OPS/FORTRAN interface was a concern as were the licensing issues involved with using a commercial tool to develop an expert system that was to be embedded in an existing FORTRAN program and distributed to engineering environments where DEC OPS would most likely not be available.

During this time vendors began offering a few tools that could access non-Lisp environments. This capability was made possible because the tools were not written in Lisp, but in more conventional programming languages like C. These included S.1, M.1, various versions of OPS, and CLIPS[1]. Of these CLIPS(C Language Integrated Production System) provided the most benefits for the project. The benefits of CLIPS were that it was not a commercial tool, it was developed at Johnson Space Center and was readily available to other NASA projects, it was written in C and thus highly portable, it could easily integrate into the FORTRAN environment, and access was available to both the source code and the people who wrote it.

3.2 The Selected Approach - CLIPS and FORTRAN

Some preliminary implementations, tests, and experiments were run to test the integration of CLIPS with FORTRAN and to determine the amount of effort required to convert the now quite large prototype of NESSUS/EXPERT from OPS to CLIPS. The results were very positive. Therefore, if NESSUS/EXPERT were implemented in CLIPS the sole remaining deficiency would be the requirement for delivery in FORTRAN. Since the primary reason for requiring delivery in FORTRAN had been to insure portability and availability to NASA, the NASA sponsor of the PSAM project consented to delivery in CLIPS. Therefore, CLIPS was selected as the implementation language for NESSUS/EXPERT.

In light of the FORTRAN interfacing capabilities that CLIPS provides, a reassessment of the NESSUS/EXPERT design was made. In previous implementations all the required NESSUS/EXPERT functions had to be implemented with the rule-based tool, including those tasks that were best suited for conventional programming methods. With CLIPS' ability to integrate with FORTRAN, an investigation was made into whether the non-AI type tasks should be moved from the rules out to FORTRAN routines.

The NESSUS/EXPERT functions were divided into two classes: functions that performed high-level tasks that are well-suited to AI solutions, such as decision making, consistency checking and evaluation and functions that performed lower-level functions which could run more efficiently implemented in FORTRAN.

The NESSUS/EXPERT tasks identified as best suited for CLIPS implementation include consistency and completeness checking, inference of data from other available data, interpretation and analysis of data, providing guidance and advice on the use of the system, and control tasks, such as running the dynamic menu system for the user interface. These tasks require either heuristic-type knowledge or knowledge concerning the relationships between the categories of NESSUS data and are well suited to implementation as forward chaining rule bases.

The NESSUS/EXPERT tasks identified as best suited for FORTRAN implementation include user I/O, file I/O, editor-type tasks, such as type and range checking, mathematical computations, and data deck generation. It is worthy of note that although the functions selected for FORTRAN implementation can execute more efficiently written in FORTRAN, the development of FORTRAN code for these functions required considerably more time than it had with the expert system building tool. These FORTRAN routines are also much less flexible than the corresponding OPS5 rules which performed basically the same functions.

The resulting system brings the best of both conventional and AI programming techniques together. CLIPS rules drive the NESSUS/EXPERT system. The CLIPS rules invoke the appropriate FORTRAN modules to perform low-level functions as needed. The CLIPS rules pass data to C interface routines which pass the data on to the FORTRAN modules as calling parameters. The FORTRAN modules make assertions into the CLIPS world via a handful of FORTRAN and C interface routines. The interactions between the CLIPS rules and the FORTRAN modules are represented in Figure 1.

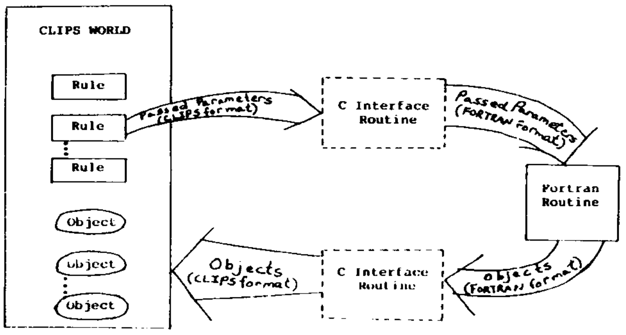

FIGURE 1. CLIPS/FORTRAN COMMUNICATION

NESSUS/EXPERT communicates with the user through both the CLIPS rules and the FORTRAN modules. The design of the interface is such that it is not apparent to the user whether CLIPS rules or FORTRAN code is driving it at any given time. The FORTRAN modules provide the interface to the NESSUS modules and the user's data files. Figure 2 shows the external interfaces to NESSUS/EXPERT.

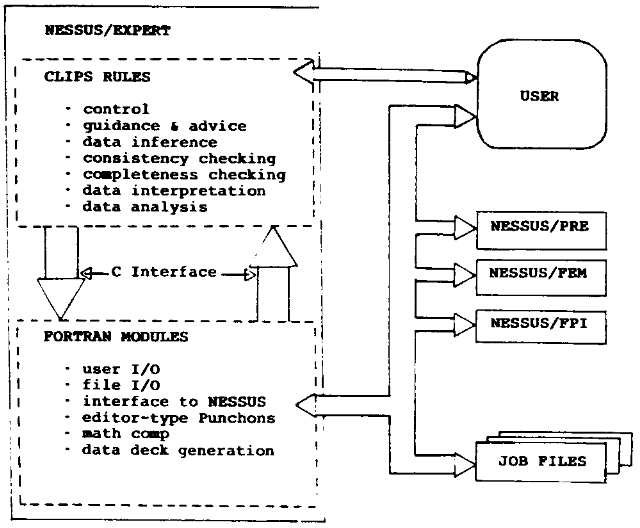

FIGURE 2. NESSUS/EXPERT EXTERNAL COMMUNICATION

4 OVERVIEW OF NESSUS/EXPERT

Currently the system consists of around 700 CLIPS rules and 300 FORTRAN routines. It runs on a VAX under VMS. A partial example session for building a deterministic data deck is presented in the appendix.

The marriage of CLIPS and FORTRAN provided several capabilities in terms of satisfying the operational requirements of NESSUS/EXPERT. The efficiency of the system is optimised because each of the languages performs the types of tasks which they do best. Well organised rule sets and FORTRAN modules provide modularity. The use of CLIPS and FORTRAN77 provides for both portability and availability to the engineering community. Thus the operational requirements of the system were satisfied.

Many of the functional requirements are satisfied via a "smart", CLIPS controlled menu system which is the basis of the user interface. Knowledge of the process that the engineer must follow to use NESSUS is encoded into the menu system. Figure 3 shows the heirarchical structure of the menus in the NESSUS/EXPERT system.

In a typical session, NESSUS/EXPERT would first present the 'JOB SELECTION' menu. The job selected specifies which structural analysis problem the user would like to work on. The user may select to continue work on a job that has been previously worked on or enter a new jobname to begin work on a new structural analysis problem. When the job is specified, NESSUS/EXPERT automatically retrieves all of the information that has previously been entered for that job. Once all of the information for the selected job has been retrieved, NESSUS/EXPERT presents the user with the 'ACTIVITY SELECTION' menu. Like most of the NESSUS/EXPERT menus, the contents of this menu are dynamically determined by NESSUS/EXPERT based on the current status of the job. For example, the activity selection menu for a job will include the option to run a deterministic analysis only if a complete and consistent deterministic datadeck exists for the job. From the 'ACTIVITY SELECTION' menu the user can move throughout the menu system as desired. The only restriction on movement through the menu system is that NESSUS/EXPERT will not provide access to menus which do not make sense in the current context.

ORIGINAL PAGE IS
OF POOR QUALITY

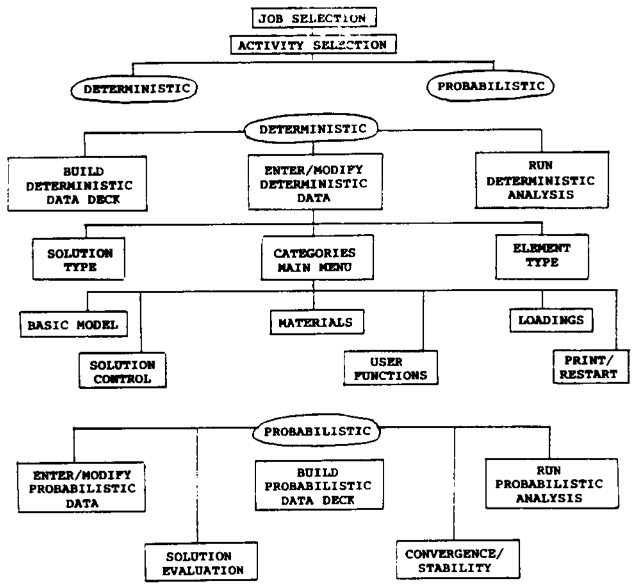

FIGURE 3. NESSUS/EXPERT MENU STRUCTURE

Within the NESSUS/EXPERT menu system, there are five basic types of screens. These five types of screens are used for five different types of interactions with the user. Consistent formats have been defined for each of the five screen types to enhance user-friendliness. These five screen types include a menu screen, a manual entry screen, a flag entry screen, a help screen, and an information screen.

The menu screen type is used by NESSUS/EXPERT to prompt the user for a choice between several available activity options. The CLIPS rules control when the menu screen appears and what items appear on it. The FORTRAN modules operate the actual input and output for the screen. Figure 4 exemplifies the NESSUS/EXPERT menu screen. The following features are common to all menu screens.

- A title on the top line indicates both the current orientation within the NESSUS/EXPERT system and the purpose of the screen.
- The name of the current job appears on the top line next to the title.
- A brief set of instructions appears after the title line.
- A HELP option is provided as a menu item. When this option is selected, NESSUS/EXPERT displays one or more screens of text which explain the current menu. When the user exits the HELP screen, NESSUS/EXPERT returns to the menu from which the HELP option was selected. This provides an on-line help facility which is context sensitive. The types of knowledge available via this help facility include knowledge about the use of NESSUS/EXPERT and knowledge about NESSUS. It serves as an enhanced, automated NESSUS user's manual.
- A QUIT option is provided as a menu item. Selection of this option usually returns the user to the next level up in the menu heirarchy.

Manual entry screens provide an editor-like means for the user to enter data directly into the NESSUS/EXPERT system with insurance that the data meets the requirements of the NESSUS modules. The manual entry screens allow the user to enter, delete, modify and view data. Manual entry screens are run by FORTRAN modules. They are invoked by CLIPS rules at the appropriate times. The FORTRAN manual entry modules contain specific

```
     ***    BASIC MODEL MENU FOR EXAMPLE     ***

SELECT THE CATEGORY OF DATA TO BE ENTERED.
(Categories marked with • have been at least partially
entered.)

  1    QUIT               2   *COORDINATES        3   *ELEMENTS

  4    BOUNDARY           5    TRANSFORMATIONS    6    TYING

  7    SPRINGS            8    MASSES             9    DASHPOTS

 10    DUPLICATENODE     11    EMBED             12   HELP

    >> 2
```

FIGURE 4. TYPICAL NESSUS/EXPERT MENU SCREEN

knowledge about types and ranges required for each piece of data within each category. Where these types and ranges vary depending on other data values that have been entered, the invoking CLIPS rules inform the FORTRAN modules of the appropriate restrictions. Figure 5 exemplifies the NESSUS/EXPERT manual entry screen. The following features are common to all manual entry screens.

- A title on the top line indicates the category and the job name.
- The most recently manipulated lines of data are displayed.
- A brief set of instructions follow the data display.
- The user can return to the previous menu by tying 'q'.
- The help facility (as described above) can be accessed by typing 'h'.
- A line of data can be deleted by typing 'd' and the line number.
- A line of data can be viewed by typing the line number.
- A line of data can be changed by simply reentering the line.

```
***    COORDINATES MANUAL ENTRY FOR EXAMPLE      ***
Coordinates data for line    1 and subsequent lines:

  1)    1     0.0000     0.0000     0.0000     0.1000
  2)    2     1.0000     0.0000     0.0000     0.1000
  3)    3     2.0000     0.0000     0.0000     0.1000
  4)    4     3.0000     0.0000     0.0000     0.1000
  5)    6     5.0000     0.0000     0.0000     0.1000
  6)    7     6.0000     0.0000     0.0000     0.1000
  7)    8     7.0000     0.0000     0.0000     0.1000
  8)    9     8.0000     0.0000     0.0000     0.1000

Enter an existing node number and  4 new data values,
new node number and  4 data values or a
line number (to view existing coordinates data).

(Enter Q(uit) when all COORDINATES data has been entered.)
 -3 1 2 3 4
```

FIGURE 5. TYPICAL NESSUS/EXPERT MANUAL ENTRY SCREEN

Flag screens provide a method of manual entry for information which NESSUS uses that does not involve data values. The manual entry screen allows pieces of information to be toggled on and off. The CLIPS rules control when the flag screens appear and the FORTRAN modules perform the actual input and output for the screen. Figure 6 exemplifies the NESSUS/EXPERT flag screen. The following features are common to all flag screens.

- A title on the top line indicates the category and the job name.
- A brief set of instructions follows the title line
- The first option (quit) allows the user to leave the flag screen.
- The second option (on) allows for the flag to be activated.
- The third option (off) allows for the flag to be deactivated.
- The fourth option (help) provides access to the help facility as described above.

```
     ***   SET/RESET FLAG FRONTALSOLUTION

     Flag is currently SET OFF.

  1  Quit

  2  SET flag to ON

  3  RESET flag to OFF

  4  HELP

     >> 2
```

FIGURE 6 - TYPICAL NESSUS/EXPERT FLAG SCREEN

The help screen is used by NESSUS/EXPERT to display text explanations to the user. The contents of the help screen will apply specifically to the user's current position within the menu system and to the current status of the job. The types of knowledge available via this help facility include knowledge about the use of NESSUS/EXPERT and knowledge about NESSUS. It serves as an enhanced, automated NESSUS user's manual. Help screens explain both what is expected of the user under the current circumstances and additional information concerning the technical aspects of the subject at hand. This provides an on-line help facility which is context sensitive. Help screens are provided by the FORTRAN modules. When the user exits the HELP screen, NESSUS/EXPERT returns to the screen from which the HELP option was selected. Figure 7 exemplifies the NESSUS/EXPERT help screen. The following features are common to all help screens.

- A title on the top line indicates the topic of help and the job name.
- Striking the return key will either display more help if available or will return the user to the previous screen.

```
                 *** ELEMENT TYPE HELP SCREEN  ***

Select the element type to be used in the analysis by
entering the number which corresponds to that element.
The element types currently available are:

  3: 4-node plane-stress. Like MARC element type 3.
  7: 8-node solid. Like MARC element type 7.
 10: 4-node axisymmetric solid. Like MARC element type 10.
 11: 4-node plane strain. Like MARC element type 11.
 75: 4-node thick/thin shell. Like MARC element type 75.
 98: 2-node Timoshenko beam. Like MARC element type 98.
151: 4-node assumed strain plane stress.
152: 4-node assumed strain plane strain.
153: 4-node assumed strain axisymmetric.
154: 8-node assumed strain solid.

The current version of MHOST only allows finite element
models consisting of a single element type.

(RETURN to continue)
```

FIGURE 7. TYPICAL NESSUS/EXPERT HELP SCREEN

The information screens provide high-level information to the user, such as warnings and advice. For example, NESSUS/EXPERT would use an information screen to alert the user when an inconsistency is detected in the model data. Figure 8 exemplifies the NESSUS/EXPERT information screen. The following features are common to all information screens.

- A title on the top line indicates the type of information involved and the job name.
- The information itself is displayed as the body of the screen.
- A list of the options available to the user in response to the given information is presented after the information.

Thus, NESSUS/EXPERT appears to the user as a set of highly flexible menu interfaces. In general, the appearance of a menu and the options available on it are determined by rules written in CLIPS that utilize knowledge about the current state of the job and what the user wishes to do. CLIPS passes any directions concerning what should be displayed to the appropriate FORTRAN routines. The

```
***             CONSISTENCY CHECK FOR EXAMPLE              ***

Element data references the following undefined nodes.

  5

TYPE  C  to enter coordinate data for the node(s) OR
      E  to change the element definition(s).
      I  to ignore the inconsistency for now.

>> i
```

FIGURE 8 - TYPICAL NESSUS/EXPERT INFORMATION SCREEN

FORTRAN routines handle the actual display of the menu and the acceptance of the input from the user. The FORTRAN routines then return any information that is needed for higher level decisions back to the CLIPS environment.

5 CONCLUSIONS

Over the past few years aritifical intelligence, especially expert system technology, has been applied to a wide variety of real world problems. At the same time, it has become apparent that in order for such systems to be truly useful, they must be able to integrate with other real world computer software systems. The level of integration required by any given system depends on its needs. The need to access certain data, either from a flat file or database, is fairly simple and straightforward. The need to share control and functionality, on the other hand, is not so easy.

NESSUS/EXPERT must not only access data, it must access and analyze a large amount of data in real time. The analysis that must be performed usually depends not on the data itself, but on certain attributes of the data, such as the number of parameters or the maximum value of a certain parameter. Processing the data and calculating these attributes is more efficiently handled by a conventional language such as FORTRAN. Analyzing these attributes and developing conclusions is better handled by a higher level language such as CLIPS.

Aside from the issue of data access, NESSUS/EXPERT must also be capable of invoking the NESSUS code itself, thus serving as a user interface to the structural analysis system. Also, due to the need to divide the tasks between FORTRAN and CLIPS based on functionlaity, NESSUS/EXPERT must be able to invoke both FORTRAN and CLIPS routines in as seamless a manner as possible. Therefore, NESSUS/EXPERT must manage control and functionality between the FORTRAN and CLIPS environments.

We have been successful in achieving the requirements for integration between an artificial intelligence and an engineering environment in NESSUS/EXPERT for several reasons. First, an appropriate division of functional tasks was defined. These tasks were then associated with the proper method for implementation, artificial intelligence or conventional. Based on the amount and type of integration required as a result of assigning tasks to implementation approaches and the other requirements of the project, an appropriate set of tools was selected, in this case CLIPS and FORTRAN. Finally, the actual integration routines were designed and implemented to insure as seamless a transition between the two environments as possible. In this way a system that takes advantage of the best in both the artificial intelligeence and engineering environments can be successfully developed in an efficient and effective manner.

6 REFERENCES

[1] Culbert, Chris, "CLIPS Reference Manual - Version 4.0," Mission Support Directorate, MPAD, NASA-JSC, March 1987.

ORIGINAL PAGE IS
OF POOR QUALITY

[2] Nakazawa, Dias, Nagtegaal, and Wertheimer, "MHOST Users' Manual - Version 3.3," April 1986.

APPENDIX - A SAMPLE SESSION

The first screen displayed by the NESSUS/EXPERT system is the JOB SELECTION menu (Screen 1). This menu allows the user to select from the jobs that have been previously worked on via the NESSUS/EXPERT system or to enter the name of a new job. Notice in Screen 1 a new job called EXAMPLE has been selected rather than selecting an already existing job.

Next NESSUS/EXPERT prompts the user for a comment that will be placed at the top of the data file for future reference. Entry of the comment is optional and is provided only for the user's convenience. Screen 2 shows the COMMENT ENTRY screen.

In most cases, NESSUS/EXPERT allows the user to determine the order in which various types of data are entered. However, there are two pieces of data that NESSUS/EXPERT requires to be entered before all others. These are the analysis type and the element type. NESSUS/EXPERT requires the entry of these before all other data due to the high degree of dependencies of the subsequent data requirements imposed by these two pieces of data. In Screen 3 STATIC is chosen for the analysis type.

In Screen 4, the element type is prompted for and the HELP option has been selected. This results in the display of a help screen for selection of the NESSUS element type (Screen 4).

In this sample session, the return key was hit to return to the element type selection menu and element type "75" was selected. Thus with the mandatory pieces of data having been entered, the SYSTEM MENU for the job is displayed (Screen 6). The options in this menu are generated dynamically based on the current state of the job. Notice there are only three options available to the user at this point. The user must enter the minimum data requirements for the deterministic datadeck before other options will be offered. This is simply because no other options make sense until the deterministic data has been entered. For this sample session, the option to enter the deterministic model data is selected.

There are around 70 categories of deterministic data that can be entered via NESSUS/EXPERT. Therefore, these categories have been organized into 6 seperate menus for ease of use. The menu shown in Screen 7 prompts the user for selection of one of these 6 menus. For our sample session the BASIC MODEL MENU has been selected.

Screen 8 shows the categories offered by the BASIC MODEL MENU. In this screen, the user has selected to enter the deterministic data for the COORDINATES category.

Most category data can be entered either from an external file or by manual entry. *(If the category was previously entered, it does not* have to be reentered. The system will automatically read in all data

```
                 ***   JOB SELECTION   ***

   SELECT THE JOB YOU WOULD LIKE TO WORK ON.
   (To begin work on a new job, enter the new jobname.)
   (to delete a job, type "D <return>" or "d <return>".)

1  QUIT           2  FKP           3  SAMPLE

4 VALID CASE3     5  HELP

   >> EXAMPLE
                         SCREEN 1
```

```
         ***   COMMENT ENTRY FOR EXAMPLE   ***

ENTER THE COMMENT FOR THE JOB.
(The comment must be strictly alphanumeric;
 no special characters.)

 Current comment:


 >> Example job for demonstration purposes
                         SCREEN 2
```

```
  ***   ANALYSIS TYPE SELECTION FOR EXAMPLE   ***

  SELECT THE ANALYSIS TYPE YOU WOULD LIKE MODEL.


1  static                     2  dynamic

3  frequency                  4  buckling

5  modal                      6  frequency domain dynamic

7  HELP

  >> 1
                         SCREEN 3
```

```
  ***   ELEMENT TYPE MANUAL ENTRY FOR EXAMPLE   ***

  Enter the element type.


1  QUIT    2  "3"    3  "7"     4  "10"     5  "11"

6  "75"    7  "98"   8  "151"   9  "152"   100  "153"

11  "154"  12  HELP

  >> 12
                         SCREEN 4
```

```
             *** ELEMENT TYPE HELP SCREEN ***

Select the element type to be used in the analysis by
entering the number which corresponds to that element.
The element types currently available are:

  3: 4-node plane-stress. Like MARC element type 3.
  7: 8-node solid. Like MARC element type 7.
 10: 4-node axisymmetric solid. Like MARC element type 10.
 11: 4-node plane strain. Like MARC element type 11.
 75: 4-node thick/thin shell. Like MARC element type 75.
 98: 2-node Timoshenko beam. Like MARC element type 98.
151: 4-node assumed strain plane stress.
152: 4-node assumed strain plane strain.
153: 4-node assumed strain axisymmetric.
154: 8-node assumed strain solid.

The current version of MHOST only allows finite element
models consisting of a single element type.

(RETURN to continue)
                         SCREEN 5
```

```
             ***   SYSTEM MENU FOR EXAMPLE   ***

  SELECT AN ACTIVITY.


1  Exit the PSAM Expert system.

2  Return to JOB SELECTION MENU.

3  Enter/Alter/Complete deterministic model data.

4  HELP

  >> 3
                         SCREEN 6
```

```
  ***   MODEL DATA GROUPS MENU FOR EXAMPLE   ***

 SELECT THE DESIRED MODEL DATA MENU.


1  BASIC MODEL MENU          2  MATERIAL DATA MENU

3  LOADINGS MENU             4  USER FILE SPECS MENU

5  SOLUTION CONTROL MENU     6  PRINTOUT CONTROL MENU

7  Return to SYSTEM MENU     8  HELP

 >> 1
                         SCREEN 7
```

associated with a job when the job is selected.) In Screen 9, the user has specified that the data be read in from a file called COORD.DAT.

When the coordinates data has been read in the system returns to the BASIC MODEL MENU for the next selection. Notice in Screen 10 that the COORDINATES option has been marked with an asterisk to indicate that this data has been entered. The elements data has also been entered in the same way although for the sake of brevity this process was not shown here. In Screen 10, the user is reselecting the COORDINATES option.

Since some COORDINATES data has already been entered, the system takes the user directly to manual entry. In Screen 11, the user has entered a bad data value (an illegal node number) for new coordinates data.

Screen 12 shows an error message resulting from the entry in Screen 10. The user now has entered "d 5" to tell the system to delete the 5th line of data.

Screen 13 shows that the data from line 5 was deleted and the remaining data was shifted to fill that line. The user has now selected to QUIT from COORDINATES manual entry.

Once the COORDINATES manual entry is left, the CLIPS rules detect that there is an unusual situation. Screen 14 shows an information screen which alerts the user to the fact that the nodes defined are not in the expected consecutive sequence. In this case, the user assures the system that this was the intention and the system forgets the matter. Had the user answered NO to the query, the system would have returned the user to manual entry for correction of the problem.

Screen 15 shows a second information screen presented by the CLIPS rules. Here the system has detected an unacceptable inconsistency between the ELEMENTS data and the COORDINATES data. Selection of either the C or E option would take the user directly to manual entry for the data type. In the example the user has selected to ignore the inconsistency for the time being. The NESSUS/EXPERT system will not forget that this inconsistency exists, however. The user will be reminded of it before a deterministic data deck is built.

Again, for the sake of brevity, screens are not provided for the entry of the remaining required deterministic data. In Screen 16, the user has moved to the SOLUTION CONTROL MENU. The FRONTALSOLUTION option has been chosen. Notice that there are 13 categories of data in this menu and none have been previously selected (no asterisks).

The FRONTALSOLUTION category is simply a flag to NESSUS, therefore no data is required. The category is either present or not

```
     ***   BASIC MODEL MENU FOR EXAMPLE     ***

   SELECT THE CATEGORY OF DATA TO BE ENTERED.
   (Categories marked with • have been at least
    partially entered.)

 1   QUIT              2   COORDINATES       3   ELEMENTS

 4   BOUNDARY          5   TRANSFORMATIONS   6   TYING

 7   SPRINGS           8   MASSES            9   DASHPOTS

10   DUPLICATENODE    11   EMBED            12   HELP

   >> 2
```
SCREEN 8

```
   *** COORDINATES ENTRY METHOD MENU FOR EXAMPLE ***

   SELECT A DATA ENTRY METHOD.

 1  QUIT

 2  Enter data by hand.

 3  Read data in from file. (Enter filename)

 4  HELP

   >> coord.dat
```
SCREEN 9

```
    ***   BASIC MODEL MENU FOR EXAMPLE     ***

   SELECT THE CATEGORY OF DATA TO BE ENTERED.
   (Categories marked with • have been at least
    partially entered.)

 1   QUIT              2  *COORDINATES       3  *ELEMENTS

 4   BOUNDARY          5   TRANSFORMATIONS   6   TYING

 7   SPRINGS           8   MASSES            9   DASHPOTS

10   DUPLICATENODE    11   EMBED            12   HELP

   >> 2
```
SCREEN 10

```
***   COORDINATES MANUAL ENTRY FOR EXAMPLE     ***

Coordinates data for line    1 and subsequent lines:

    1)    1     0.0000     0.0000     0.0000     0.1000
    2)    2     1.0000     0.0000     0.0000     0.1000
    3)    3     2.0000     0.0000     0.0000     0.1000
    4)    4     3.0000     0.0000     0.0000     0.1000
    5)    6     5.0000     0.0000     0.0000     0.1000
    6)    7     6.0000     0.0000     0.0000     0.1000
    7)    8     7.0000     0.0000     0.0000     0.1000
    8)    9     8.0000     0.0000     0.0000     0.1000

Enter an existing node number and  4 new data values,
new node number and  4 data values or a
line number (to view existing coordinates data).

(Enter Q(uit) when all COORDINATES data has been entered.)

 -3 1 2 3 4
```
SCREEN 11

```
***   COORDINATES MANUAL ENTRY FOR EXAMPLE     ***

INVALID NODE NUMBER ENTERED.
LEGAL RANGE: 1 TO  400.

Coordinates data for line    1 and subsequent lines:

    1)    1     0.0000     0.0000     0.0000     0.1000
    2)    2     1.0000     0.0000     0.0000     0.1000
    3)    3     2.0000     0.0000     0.0000     0.1000
    4)    4     3.0000     0.0000     0.0000     0.1000
    5)    6     5.0000     0.0000     0.0000     0.1000
    6)    7     6.0000     0.0000     0.0000     0.1000
    7)    8     7.0000     0.0000     0.0000     0.1000
    8)    9     8.0000     0.0000     0.0000     0.1000

Enter an existing node number and  4 new data values,
new node number and  4 data values or a
line number (to view existing coordinates data).

(Enter Q(uit) when all COORDINATES data has been entered.)
 d 5
```
SCREEN 12

```
***   COORDINATES MANUAL ENTRY FOR EXAMPLE     ***

Coordinates data for line 5 and subsequent lines:

    5)    6     5.0000     0.0000     0.0000     0.1000
    6)    7     6.0000     0.0000     0.0000     0.1000
    7)    8     7.0000     0.0000     0.0000     0.1000
    8)    9     8.0000     0.0000     0.0000     0.1000
    9)   10     9.0000     0.0000     0.0000     0.1000
   10)   11    10.0000     0.0000     0.0000     0.1000
   11)   12     0.0000     1.0000     0.0000     0.1000
   12)   13     1.0000     1.0000     0.0000     0.1000

Enter an existing node number and  4 new data values,
new node number and  4 data values or a
line number (to view existing coordinates data).

(Enter Q(uit) when all COORDINATES data has been entered.)

 q
```
SCREEN 13

ORIGINAL PAGE IS
OF POOR QUALITY

present. Screens 17 and 18 demonstrate how NESSUS/EXPERT provides for this type of category to be turned on and off. In the example, the category is turned on.

Screen 19 shows the SOLUTION CONTROL MENU consequent to the FRONTALSOLUTION option being selected. Notice that there are now only 12 categories of data in this menu. NESSUS/EXPERT has removed the BANDMATRIX option from the menu. This is because FRONTALSOLUTION and BANDMATRIX are mutually exclusive options for the NESSUS input.

Screen 20, the final screen in this example shows that once the minimum data requirements for the deterministic model have been entered, options are added to the system menu accordingly.

```
***   COORDINATES MANUAL ENTRY FOR EXAMPLE     ***
 Undefined COORDINATES NODES:
5
Should these NODES be left undefined (Enter Y or N).
 y
```

SCREEN 14

```
***          CONSISTENCY CHECK FOR EXAMPLE              ***
 Element data references the following undefined nodes.
   5
 TYPE  C  to enter coordinate data for the node(s) OR
       E  to change the element definition(s).
       I  to ignore the inconsistency for now.
>> i
```

SCREEN 15

```
    ***   SOLUTION CONTROL MENU FOR EXAMPLE      ***
   SELECT THE CATEGORY OF DATA TO BE ENTERED.
   (Categories marked with * have been at least
    partially entered.)
 1    QUIT                      2    ITERATIONS
 3    INCREMENTS                4    TIME
 5    BANDMATRIX                6    BFGS
 7    DEFORMATION               8    DISPLACEMENTMETHOD
 9    FRONTALSOLUTION          10    LINESEARCH
11    LOUBIGNAC                12    OPTIMIZE
13    SECANTNEWTON             14    TANGENT
15   HELP
>> 9
```

SCREEN 16

```
    ***   SET/RESET FLAG FRONTALSOLUTION
   Flag is currently SET OFF.
 1  Quit
 2  SET flag to ON
 3  RESET flag to OFF
 4  HELP
   >> 2
```

SCREEN 17

```
   ***   SET/RESET FLAG FRONTALSOLUTION
  Flag is currently SET ON.
 1  Quit
 2  SET flag to ON
 3  RESET flag to OFF
 4  HELP
   >> 1
```

SCREEN 18

```
   ***   SOLUTION CONTROL MENU FOR EXAMPLE     ***
   SELECT THE CATEGORY OF DATA TO BE ENTERED.
   (Categories marked with * have been at least
    partially entered.)
 1    QUIT                      2    ITERATIONS
 3    INCREMENTS                4    TIME
 5    BFGS                      6    DEFORMATION
 7    DISPLACEMENTMETHOD        8   *FRONTALSOLUTION
 9    LINESEARCH               10    LOUBIGNAC
11    OPTIMIZE                 12    SECANTNEWTON
13    TANGENT                  14   HELP
   >> 1
```

SCREEN 19

```
            ***   SYSTEM MENU FOR EXAMPLE     ***
   SELECT AN ACTIVITY.
1  Exit the PSAM Expert system.
2  Return to JOB SELECTION MENU.
3  Change job comment.
4  Reselect analysis type.
5  Enter/Alter/Complete deterministic model data.
6  Enter/Alter/Complete random variables.
7  Build Deterministic Datadeck.
8  Run deterministic analysis.
9  HELP
   >> 6
```

SCREEN 20