N92-16592

CLIPS APPLICATION USER INTERFACE FOR THE PC

Jim Jenkins, Rebecca Holbrook, Mark Shewhart, Joey Crouse, and Capt Stuart Yarost

Air Force Logistics Command (AFLC)
Acquisition Logistics Division, Joint Technology Applications Office (ALD/JTI)
Artificial Intelligence Support Center (AISC)
Wright Patterson Air Force Base, Ohio 45433

Abstract. The majority of applications that utilize expert system development programs for their knowledge representation and inferencing capability require some form of interface with the end user. This interface is more than likely an interaction through the computer screen. When building an application the user interface can prove to be the most difficult and time consuming aspect to program. Commercial products currently exist which address this issue. To keep pace CLIPS will need to find a solution for their lack of an easy-to-use Application User Interface (AUI). This paper represents a survey of the DoD CLIPS' user community and provides the backbone of a possible solution.

INTRODUCTION

A transparent, easy to use, and more visually pleasing end-user interface for applications developed in CLIPS is needed. Bernard Engel of Purdue University clearly presented this point in the paper he gave at the First CLIPS Conference Proceedings concerning their in-house expert system development. Reference [1]. In addition, McDonnell Douglas' MCAIR Artificial Intelligence Center is developing a similar product. Reference [2]. At MCAIR, they developed the CLIPS Advanced Development Enhancement (CADE) as part of the Technical Expert Aircraft Maintenance System (TEAMS) project. At the same time we were developing our own enhancements to CLIPS and researched the above contacts concerning the simililarities and differences. Given this activity and our own in-house requirement for a more palatable user interface, we established the following initial requirements for a CLIPS AUI:

(1) The interface should consist of simple function calls from within CLIPS.

(2) The interface should not over burden the CLIPS applications programmer with excessive control of the interface screens.

(3) The interface must provide easy to use window-like Input and Output functions with the following minimal capabilities:

(a) Display blocks of text.

(b) Allow users to input single or multiple responses to multiple choice questions via arrow keys.

(c) Obtain user input through the keyboard.

(4) The text, questions, and menu options associated with an application can be read from a flat ASCII text file. That is, the text is neither compiled nor entered as part of the CLIPS knowledge base. The CLIPS applications programmer must have the ability to use multiple ASCII files for this purpose.

The prototype of the CLIPS AUI developed at Purdue University meets requirements 1 through 3 above. However, the Purdue implementation places all the text within the CLIPS knowledge base. We found that separating the text from the knowledge base allows easy review/editing by the expert and improves memory management. The product developed by MCAIR meets all of the above requirements and includes some additional features such as rule-chaining and a pseudo-nonmonotonic reasoning capability. Our own in-house product, INTCLIPS, also meets all of the above requirements and was developed for use on current and future projects. All these efforts address the same AUI issues and approached the problem in similar ways. To avoid further duplication of effort, and to support the CLIPS community, we present this paper.

This paper provides a technical survey of the user requirements for a more effective application user interface to CLIPS. To begin with, a detailed description of the AUI prototype, INTCLIPS, is presented. Second, an outline of the AUI customer feedback is given followed by the technical issues concerning the implementation issues of the feedback. Finally, a survey is presented of the commercial products which may provide an alternative solution for building the end-user interface.

AUI PROTOTYPE

We have developed a set of AUI functions that can be embedded within the CLIPS source code. These functions provide the developer of an expert system application with the basics required for an AUI, thereby facilitating the process of rapid prototyping and development. We designed this set of functions to provide a friendly window-like environment for the end user. The basis of the AUI is the capability to retrieve data from either a standard ASCII text file or the keyboard and the capability to display data to the terminal.

The function **write_values** is used to access and display the contents of the facts list. The following will describe in detail the utility of each function.

Init_clips_display

The user defined function, init_clips_display, is used to initialize each paragraph and question display routine. The function call is required to initialize the pointers to the flat ASCII files which contain the questions and text. See **Table 1** for an example.

Usage : (**init_clips_display** *arg_1 arg_2 arg_3*)

arg_1 : Name of the flat ASCII file that contains the questions or text. Arg_1 may not be omitted.

arg_2 : Name of the file that will be created to hold the index information for the flat ASCII file named in arg_1. Arg_2 may not be omitted.

arg_3 : Either "paragraph" or "question" depending upon the contents of arg_1. ARG_3 may not be omitted.

```
(defrule Initialize
   (initial-fact)
=>
   (init_clips_display "pgh.txt" "pgh.idx" "paragraph")
   (init_clips_display "ques.txt" "ques.idx" "question")
   (banner "on" "Engine Diagnostic Expert System")
   (assert (ready)))
```

Table 1. Example of Rule Using Init_clips_display and Banner

Clear_screen

The user defined function, clear_screen, is used to clear the contents of the video screen. See **Table 2** for an example.

Usage : (**clear_screen**)

Banner

The user defined function, banner, is used for aesthetic purposes to add a title or banner to the expert system screen. See **Table 1** for an example.

Usage : (**banner** *arg_1 arg_2*)

arg_1 : The argument takes on two values either "on" or "off". Arg_1 may not be omitted.

arg_2 : The title which is to be placed within the banner at the top of the screen (ie. "My program title"). Arg_2 may be omitted.

Write_paragraph

The user defined output function, write_paragraph, is used to display a paragraph of text within a specified paragraph block. This function uses formatted or unformatted paragraphs and displays a block of text associated with a key-word on the screen in a window-like format. See **Table 2** for an example.

Usage: (**write_paragraph** *arg_1 arg_2*)

arg_1 : Name of the file that holds the index information for the flat ASCII file which contains the textual information associated with the key-word in arg_2. Arg_1 may not be omitted.

arg_2 : The key-word or "paragraph name" which identifies the block of text to be retrieved. Arg_2 may not be omitted.

```
(defrule print-repair ""
  (list-repairs)
  (name name)
   =>
  (clear_screen)
  (write_values "pgh.idx" "repair_title" "repair" ?name)
  (clear_screen)
  (write_paragraph "pgh.idx" "final_statement" ?name)
  (system "cls")
  (exit))
```

Table 2. Example of Rule Using Clear_screen, Write_paragraph, and Write_values

Ask_menu_question

The user defined input function, ask_menu_question, is used when a question is asked and multiple responses are presented in a menu selection block. This function displays the question text and list of possible answers for the question associated with a key-word. The user selects an answer using the arrow keys. Single and multiple answers are both permitted. See Table 3 for an example.

Usage: (**ask_menu_question** *arg_1 arg_2 arg_3 arg_4*)

arg_1 : Name of the file that holds the index information for the flat ASCII file which contains the textual information associated with the key-word in arg_2. Arg_1 may not be omitted.

arg_2 : The key-word or "question name" which identifies the block of text to be retrieved for the question. Arg_2 may not be omitted.

arg_3 : A string that is to be associated with the selected answer(s) in the CLIPS fact list. Arg_3 may not be omitted.

arg_4 : Either "single" or "multiple". "Multiple" allows one or more answers to be selected from the menu. If arg_4 is omitted, the default is "single".

```
(defrule determine-engine-state ""
   ?rem <- (query phase)
   (not (working-state engine ?))
   =>
   (retract ?rem)
   (clear_screen)
   (ask_menu_question "ques.idx" "determine_engine_state"
                      "working-state engine"))
```

Table 3. Example of Rule Using Ask_menu_question

Typed_in_question

The user defined input function, typed_in_question, is used when there is a need for input from the keyboard. The function displays the text of a question associated with a key-word and the user enters the answer from the keyboard. Only single answers are permitted. See **Table 4** for an example.

Usage : **(typed_in_question** *arg_1 arg_2 arg_3 arg_4***)**

arg_1 : Name of the file that holds the index information for the flat ASCII file which contains the textual information associated with the key-word in arg_2. Arg_1 may not be omitted.

arg_2 : The key-word or "paragraph name" which identifies the block of text to be retrieved for the question. Arg_2 may not be omitted.

arg_3 : A string that is to be associated with the selected answer(s) in the CLIPS fact list. Arg_3 may not be omitted.

arg_4 : Either "int", "float", "string", or "word". This indicates the type of answer expected. Type checking is done. If the type check fails, the user is prompted to re-enter the response. If arg_4 is omitted, the default is "string".

```
(defrule get-user-name
  (need name)
  =>
  (clear_screen)
  (typed_in_question "pgh.idx" "get-name" "name"))
```

Table 4. Example of Rule Using Typed_in_question

Write_values

The user defined function, write_values, displays the contents of the CLIPS fact list whose firs entry matches a particular attribute name. Above this attribute list a text block will display th text associated with the key-word found in the question file.

This function can include the values of CLIPS variables (string or float) within the body o a paragraph. This format option requires the inclusion of a corresponding number of additiona arguments within the function call ,write_values, to provide the ability to create formatte paragraphs. See **Table 2** for an example. **Table 5** illustrates an ASCII file that contains format character, %s, therefore the function call to write_values would require one additiona argument in order for this format character to be utilized.

Usage : (**write_values** *arg_1 arg_2 arg_3 arg_n*)

arg_1 : Name of the file that holds the index information for the flat ASCII file whic contains the textual information associated with the key-word in arg_2. Arg_1 may not b omitted.

arg_2 : The key-word or "paragraph name" which identifies the block of text to b retrieved. Arg_2 may not be omitted.

arg_3 : A string that represents an attribute name. Each element in the CLIPS fact list tha has arg_3 as its first element is displayed without that element.

arg_4 : The variable name, **?var**, which corresponds to a valid variable name defined withi the INTCLIPS knowledge base. Arg_4 through arg_n may be omitted.

arg_n : The variable name, **?var**, which corresponds to a valid variable name defined withi the INTCLIPS knowledge base. Arg_4 through arg_n may be omitted.

------ COMMENTS ----------	----- ASCII FILE ----------
Delimiter between text blocks	@
Identifier for text block	repair_title
Line #1 with format character	The repairs for %s are listed below.
Delimiter between text blocks	@

Table 5. Sample ASCII File with the Format Character, %s, Included

ASCII File Format

Table 6 and Table 7 provide examples of the INTCLIPS ASCII text files that would b used for a typical application. The format of a paragraph file is different than the format o a question file. Separating the two file formats was implemented in order to provide a more organized method of developing the knowledge base.

------ *COMMENTS* ----------	----- ASCII FILE ----------
Delimiter between text blocks	@
Identifier for text block	system_banner
line #1 for text block	The Engine Diagnostic Expert System
Delimiter between text blocks	@
Identifier for text block	ask-users-name
line #1 for text block	What is your name?
Delimiter between text blocks	@
Identifier for text block	suggested-repairs
line #1 for text block	The Engine Diagnostic Expert System
line #2 for text block	is now complete.
Delimiter between text blocks	@

TABLE 6. Sample ASCII File for Paragraph Text

--------- *COMMENTS* ----------	----- ASCII FILE -------
Delimiter between question blocks	@
Identifier for question block	determine_engine_state
Number of responses in menu	3
Response #1	normal
Response #2	unsatisfactory
Response #3	does-not-start
Line #1 for question text	What is the working state of the
Line #2 for question text	engine?
Delimiter for explanatory text	#
Line #1 of explanatory text	-- Optional Explanatory Text --
Delimiter between question blocks	@
Identifier for question block	determine_rotation_state
Number of responses in menu	2
Response #1	yes
Response #2	no
Line #1 for question text	Does the engine rotate?
Delimiter for explanatory text	#
Delimiter between question blocks	@
Identifier for question block	determine_sluggishness
Number of responses in menu	2
Response #1	yes
Response #2	no
Line #1 for question text	Is the engine sluggish?
Delimiter for explanatory text	#
Delimiter between question blocks	@

TABLE 7. Sample ASCII File for Questions

AUI CUSTOMER FEEDBACK

The CLIPS Help Desk provided the names of DoD CLIPS' users. Users requested a total of 83 copies of INTCLIPS for review. As a result of the accompanying INTCLIPS survey (see Attachment), we defined the following additional requirements for the CLIPS AUI.

(a) Allow retrieval of answers from a pre-processed ASCII file.

(b) Add the ability to size and move windows around.

(c) Make tool portable across multiple platforms.

(d) Adapt AUI to CLIPS version 5.0.

(e) Make CLIPS AUI MS-windows 3.0 compatible.

(f) Add mouse support.

(g) Add hypertext capability.

(h) Add on-screen form fill capability.

(i) Allow multiple text windows per screen.

(j) Allow scrollable text.

(k) Add graphics import capability.

(l) Add sufficient error trapping.

TECHNICAL ISSUES

We developed the AUI functions using Turbo C and embedded them within CLIPS as user defined functions that we then recompiled and renamed as INTCLIPS. The development of an expert system from within CLIPS and INTCLIPS is identical except that the knowledge base developed from within INTCLIPS now can call upon the user defined functions.

The AUI consists of a keyword based text retrieval system and a character based windowing and menu system - both of which are very basic in their underlying concepts. Certainly the functionality in our AUI prototype is basic. Our customer feedback indicates that the CLIPS user community is interested in more functionality than our AUI currently provides. Each suggested enhancement has direct impact on both the AUI developer as well as the CLIPS developer who will use the CLIPS AUI. Based upon our experience, we address the impact of several suggested AUI improvements below.

Portability of AUI Code

The only compiler specific code in the AUI involves the functions used to manipulate the screen. Generic screen manipulation functions could be used in the AUI code and would depend on designated compiler options set by the user. Such changes would have no impact upon the CLIPS application developer. Development of this capability would require familiarity with a variety of compilers and their screen manipulation or graphics functions.

Graphical Based Interface

We can address this issue the same way the portability of AUI code is addressed above. Such a change would have no impact upon the CLIPS application developer. Due to the finer resolution of the graphics screens, the construction of screens may be more tedious.

Mouse Support

Basic mouse support that replicates the functions of the arrow keys is a simple process to implement. On the other hand, advanced mouse support that allows window resizing and scrolling of text would be a more complicated matter.

CLIPS 5.0 Upgrade

Only slight modifications would be required for this implementation since only the names of a few external function calls have been modified for CLIPS version 5.0.

Ability To Move Windows

Many CLIPS AUI users would like the ability to place the windows of text anywhere on the screen as opposed to the default centered window. We can accomplish this feature by adding arguments to the CLIPS functions to indicate the location of the window. This feature would provide no major complications for the AUI developer.

Multiple Window Overlays

This capability is similar to the ability to move the location of a window. A toggle key must be present for switching back and forth from one window to the next. Mouse interaction also could accomplish toggling from one window to the next .

Scrollable Windows

Experience and customer feedback indicated that scrollable windows are a basic requirement that any AUI should meet. Scrollable windows would add no complications for the CLIPS application developer, but implementing this feature would add a bit more complexity to the AUI code.

Ability To Size Windows

An additional screen control feature that some CLIPS AUI users have requested is the ability to size windows. While on the surface this looks comparable to adding the ability to move the window about the screen, there are some fundamental issues involved when choosing to implement this option. The underlying text retrieval system implemented in the AUI prototype is WYSIWYG - What You See Is What You Get. That is, the text is displayed on the screen as it appears in the ASCII file. The implementation sizable windows would require manipulation of the text in a wrap around fashion and the use of scrollable windows. The only advantage this would provide is the ability to display the same paragraph in various sized windows.

Hypertext Capability

One key motivation behind the development of the AUI prototype was the large number of expert system applications that are "information intensive." Many applications simply ask a few questions and then display large amounts of information. The most basic method of displaying

information is the block of text in a window as implemented with the INTCLIPS function write_paragraph. Hypertext offers a more sophisticated method of information display. A very basic hypertext feature can be implemented using the existing AUI keyword text retrieval functions. In fact, the implementation would involve simple modifications to the existing write_paragraph function. Words surrounded by special symbols could be placed in the ASCII text files to be used as links to other blocks of hypertext. These words would simply be keywords to other blocks of text in that ASCII text file. In this way, the function implementing write_paragraph would be recursively called each time the user selected a highlighted "hot link" in the displayed text.

Form Fill Capability

We could develop a form fill capability with the same technique of utilizing a flat ASCII file and special characters to identify the various input sections of the form. The development could become somewhat complicated due to the variability and number of parameters that it may require to use.

Ability To Import Graphics

Some applications may require the ability to display graphic images. One such application that is being developed by the Department of the Army's Environmental Lab. Reference [3]. There project consists mainly of graphic presentations and therefore user defined graphic display functions were recompiled into an enhanced version of CLIPS named GCLIPS. The most direct way to implement this is with some of the freeware applications that can display graphic images saved in an appropriate file format. Since (1) there are many such freeware and commercial products available, (2) the implementation would involve simply a system call, and (3) CLIPS currently supports system calls, therefore by default this feature is available in the AUI prototype.

MS-Windows Interface

Several customers indicated a strong desire for MS-Windows compatibility. While MS-Windows compatibility has advantages, there are three major drawbacks to implementing such a feature: (1) A large amount of time and effort would be required, (2) the entire CLIPS application would probably have to be modified, and (3) potential users would be limited to MS-Windows users.

COMMERCIAL PRODUCTS

User Interface Management Systems

The following is a sampling of the companies that provide user interface management systems. These products are tools that can be used to develop a CLIPS AUI but involve integration issues and runtime costs. They offer a variety of functions for developing windows, menus, data entry screens, importing graphics, help functions, mouse support, scrollable windows, text editors, key input validation, etc. In addition, many of these products offer interactive screen designers and code generators. These products also support libraries of functions over a wide variety of platforms. References [4,5].

Creative Programming Consultants Inc.
Box 112097
Carrollton, Texas 75011
(214)416-6447

Product: VitaminC 3.2 and 4.0 with VCscreen 3.2

Vermont Creative Software
Pinnacle Meadows
Richford, VT 05476
1-800-848-1248
Product: Vermont Views with Designer v2.0 and GraphEx

Solution Systems
372 Washington Street
Wellesley, MA 02181
1-800-677-0001
Product: C-Worthy 2.0

Copia International LTD.
Roundhill Computer Systems
1342 Avalon Court
Wheaton, IL 60187
(706)682-8898
Product: Panel Plus II

Oakland Group Inc.
(Subsidiary of Liant Software Corp.)
Cambridge, Mass
1-800-233-3733
(617)491-7311
Product: C-scape 3.2 with Look & Feel 3.2

Expert System Development Environments

The following is a sampling of some of the companies that provide expert system development environments that assist with the development of the AUI.

Information Builders, Inc.
503 Fifth Avenue
Indialantic, Florida 32903
1-800-444-4303
Product: Level 5

Neuron Data Systems
156 University Avenue
Palo Alto, CA 94301
1-800-876-4900
Product: Nexpert Object

AI Corp, Inc.
1700 Rockville Pike
Suite 400
Rockville, MD 20852
(301)881-8100
Product: KBMS and 1st Class

CONCLUSION

After completing a comprehensive survey of the DoD CLIPS' user community we have found that there is a great need to enhance CLIPS with an AUI development tool. We recommend the results of this survey be utilized as a basis for such a tool development, and the next version of CLIPS should carefully weigh the importance of implementing such a tool. The user defined functions that we offer to share with the CLIPS community unfortunately, provide only the first step in meeting the customer needs. The more important contribution is the user contacts and the corresponding feedback acquired. Copies of INTCLIPS and reprints of this paper are available to government agencies upon request.

REFERENCES

[1] Engel, Bernard A., et al, "CLIPS Interface Development Tools and Their Applications", First CLIPS Conference Proceedings, Vol II, August 1990, p. 458 - 469.

[2] Blankenship, Keith and Reichart, Rick, McDonnell Douglas, MCAIR Artificial Intelligence Center, Mail Code: 1065205.

[3] Smith, Craig, Research Biologist, Department of the Army, Environmental Laboratory, Waterways Experiment Station, Corps of Engineers, 3909 Halls ferry Road, Vicksburg, Mississippi 39180-6199.

[4] Mirecki, Ted, "Interface Tools Smooth Transition to OS/2", PC Week Reviews, May 20, 1991, p. 125 - 129.

[5] Robie, Jonathan, "Three C Language Screen-Utility Packages for PCs", Byte, October 1987, p. 223 - 229.

CLIPS Application User Interface Feedback Questionnaire

ALD/JTI
ATTN: Rebecca Holbrook
Wright-Patterson AFB, OH 45433
DSN: 785-3303 COMM: (513) 255-3303

All CLIPS Application User Interface (AUI) testers/users:

Please complete this feedback questionnaire to provide us with an opportunity to make CLIPS AUI a better product. We hope to compile the survey results in July, so please return this questionnaire to the address above by **30 June 1991.**

NAME: _____ DATE: _____

ADDRESS: _____ COMM: (__) _____

_____ DSN: _____

MY JOB IS:
_ Engineer Scientist _ Equipment Specialist _ Senior Mngt
_ System Manager _ Item Manager _ Middle Mngt
_ Maint Technician _ Logistician _ Other:_____

PLEASE ANSWER THE FOLLOWING QUESTIONS. CIRCLE ALL APPROPRIATE ANSWERS WHERE APPLICABLE, ADDING YOUR COMMENTS WHENEVER POSSIBLE.

GENERAL:

1-1. Were you able to test the CLIPS AUI software as planned? If not, please explain.

1-2. What computer configuration did you use?

CPU:	Z-248	Desktop III	Other:_____
monitor:	EGA	VGA	Other:_____
mouse:	Yes	No	

1-3. What are your overall impressions of CLIPS AUI?

1-4. Specifically, what did you like?

1-5. Specifically, what did you dislike?

1-6. Of what relative value did you find the following:
[Range from 1 (no value) to 10 (highest value), 0 - did not use] Comments

_ CLIPS AUI User's Manual

_ CLIPS AUI Application Color Custimization Program (MODCFG.EXE)

_ CLIPS AUI Source Code

CLIPS AUI SPECIFIC:

2-1. Of what relative value did you find the following:
[Range from 1 (no value) to 10 (highest value), 0 - did not use] Comments

_ CLIPS AUI ***clear_screen*** function

_ CLIPS AUI ***write_paragraph*** function

_ CLIPS AUI ***ask_menu_question*** function

_ CLIPS AUI ***typed_in_question*** function

_ CLIPS AUI ***write_values*** function

_ CLIPS AUI ***banner*** function

2-2 Do you have any recommendations for additional CLIPS AUI functions?

2-3 What level of expertise do you have as a CLIPS programmer?

None Novice Journeyman Expert

2-4 If you were planning to develop an expert system in CLIPS, would you use CLIPS AUI?

yes no unsure

2-5 Do you intend to make run-time versions of your CLIPS applications?

yes no unsure