N92-16605

A CLIPS-BASED TOOL FOR AIRCRAFT PILOT-VEHICLE INTERFACE DESIGN*

Thomas D. Fowler California State Polytechnic University, San Luis Obispo

Steven P. Rogers, Ph.D.

Anacapa Sciences, Inc., Santa Barbara, California

Abstract

The Pilot-Vehicle Interface of modern aircraft is the cognitive, sensory, and psychomotor link between the pilot, the avionics modules, and all other systems on board the aircraft. To assist Pilot-Vehicle Interface designers a CLIPS-Based tool has been developed that allows design information to be stored in a table that can be modified by rules representing design knowledge. Developed for the Apple Macintosh®, the tool allows users without any CLIPS programming experience to form simple rules using a point-and-click interface.

INTRODUCTION

Development of the Pilot-Vehicle Interface (PVI) is one of the most challenging aspects of advanced aircraft programs. The PVI is the cognitive, sensory, and psychomotor link between the pilot, the avionics modules, and all other systems on board the aircraft. To reduce the pilot's workload while maximizing his situation awareness, the PVI must optimize the flow of information between the pilot and the hardware/software systems, respond smoothly to his immediate needs, and present the required information elements using the most readily comprehensible methods.

It is tempting to believe that advancing technology must have made the PVI of modern aircraft better than ever before. In fact, according to many pilots, the opposite is true. Although modern military aircraft have better weapons, better sensors, better computers, and better displays than in years past, the barrage of information from all the cockpit systems is nearly overwhelming, the data from various systems are not integrated to support the pilot's tasks, and the display types and formats are often ill-chosen. To acquire the information necessary for a given task, the pilot may have to extract data from several different systems and mentally translate or manipulate the data, while simultaneously dealing with the unrelenting workload imposed by aviation, navigation, and communication tasks.

The general requirements for an intelligent PVI for an advanced aircraft are that it be capable of filtering out information that is not currently useful, prioritizing the most useful information, prompting the pilot to examine the most important information, and integrating the functionally related data into an easily comprehended presentation. In order to determine the best presentation, the PVI must rapidly perform a layered series of

*This work is supported by an Army Small Business Innovation Research Contract (No. NAS213391) administered by the Aeroflightdynamics Directorate, NASA-Ames Research Center, Moffet Field, California. Dr. Edward M. Huff is the Contracting Officer's Technical Representative.

information presentation choices to select the best display modality, location, and format for the pilot's current activity.

In order to meet these general requirements for the PVI, it is necessary to base the system design on the pilot's functions and information requirements, rather than focusing on the technology of the aircraft systems. Taking such an information requirements approach, however, requires that thorough, systematic analyses be performed to identify the specific information elements necessary to perform each pilot function and to determine the various attributes of these elements. Next a set of rules must be developed to select the best display modalities, locations, and formats for integrating and presenting the information given the specific information elements and their attributes.

The necessity of a database of information elements, and rules linking the information elements to optimal presentation methods, strongly suggested that an expert system be developed to manage and guide the display method selection process. Such an expert system must be designed to organize the extensive collections of analytical information, document the controlled growth of the rules for information display, and manage the application of the rules to the database for selection of display methods in a methodical, consistent, and easily understood manner.

To assist in the development of such an expert system, a CLIPS-Based tool has been developed that allows the information elements and their attributes to be stored in a table that can be modified by rules representing design knowledge. The tool allows the user to create rules incrementally using a simple point-and-click interface. The following sections describe the tool from the user's perspective and discuss the use of CLIPS (NASA 1991) on the Macintosh for its implementation.

USER LEVEL VIEW

The tool presents the user with a table contained in a window that has horizontal and vertical scrolling capabilities. Rows and columns can be manipulated using cut, copy, and paste commands available from the menu bar. Entries in the table can be selected by clicking on them with the mouse or by using the arrow keys on the keyboard.

			PUI Teolo			
fafe_Type	Info_Etoment	×	٧	Pladality	Lecation	Fermet
Flight_Plan	Time_Estimates	Lev	Ret			
Flight_Plan	Altitutes	High	Green			
Weether	Wind_Speed	Madium	61 ye			
Weether	Visibility	High	81 us			
Aircraft	Fuet_Remaining	High	Red			
Aircraft	011_Pressure	Lev	Red			
		·				

Figure 1. Information Element Table

The first row in the table is used to identify the type, name, and attributes of each information element. Figure 1 shows a table consisting of six information elements each having the attributes X, Y, Modality, Location, and Format.

Columns entries that are set by the user are tagged as "input" columns, while those set as the result of rule execution are designated "output" columns. In figure 1, columns X and Y are input columns with entries that the user is responsible for setting. The Modality, Location, and Format columns are output columns. Associated with each column is a list of value options from which each entry is selected. Figure 2 shows the input of a list of value options for column X and its designation as an input column.



Figure 2. Column settings dialog

Once the list of value options has been set for a column, the dialog shown in Figure 3 is used to set the values of its entries. In this case, the value of the attribute X for the Wind_Speed information element is being set to Medium. The information type and attribute for the entry is indicated at the top of the dialog. If an entry already contains a value, then it is displayed in the box below the type and attribute indicators.



Figure 3. Attribute entry dialog

A list of value options must also specified for the Modality, Location, and Format columns for use by the rules which set them. Restricting the values of entries in this manner guarantees correctness of input and insures the consistent usage of symbols by user-created rules.

Adding Rules To The Table

Rules can be formed that have left-hand side (LHS) patterns that match values in input and output columns, and right-hand side (RHS) clauses that set values in output columns. Rule creation is facilitated by the dialog shown in Figure 4(a). Here, a rule with name "Rule1" has been formed that will apply to information elements that have attribute X set to Low and attribute Y set to Red. The consequence of Rule1's execution is to set the attributes Modality, Location, and Format to Visual, Display1, and Iconic, respectively. Figures 4(b) and 4(c) show the formation of additional rules.

		PUI Tabl	•			
info_Type	Info_Element X	٧	Modelity	Location	For met	
Flight_Plan	Bute Neme (Bute)		<u> </u>			
flight_Plan						
Weether	Attributes	Operato	r	Volues	#	
Weether	info_Element	≌⊛•	tow		소	
Aircraft	H	00	Medium			
Aircraft	Modelity		mgn			
	Location	1100				#
	Format					
			1			
					Q	
	Insert Clear				1	1
	IF				6	
	R - 1000					
1	Y = Red					
	i Modelitu => Nisuel					
	Location -> Display1					10 41
l	Formet -> Iconic					1
	L	<u>:</u> -			Q	
				<u></u>	ليصحد	C
2						

Figure 4(a). Rule1

		PUI Tabi	•			
lafa_Type	Info_Element X	٧	Pladelity	Lecation	fermel	
Flight_Plan	Rule Name Rule 7		Sava	Gemaile		
Flight_Plan						
Weather	Attributes	Operato	<u> </u>	Values		
weether .	info_£iement	₽ ●•	Red		엄마	
Intraft	N		breen			
Intraft	Modelitu	- o>				
	Location	ō∢			111	
	Format					
						rs And
			L		K	ĥ
	Insert Clear					182.
	IF					1
	N>- Medium					
	Y Blue					8
1						p.
	Lecation +> Display2					1
	Formet -> Numeric					E.
					Q	-15-42
L						i c
			ويتعادمون والمحادثة والمحاد	1982	257 577 308	

Figure 4(b). Rule2

ORIGINAL PAGE I OF POOR QUALIT

L-Type info_Element x Y redelity Location format Int_Plan Rute X Y redelity Location format Int_Plan Rute X Y redelity Location format Rute X Y redelity Location format form					PUI Teble				
Int_Plan Rule Name Rule 3 Save Remove Quit Rh_Plan Rttributes Operator Ualues ather Modelity O - Numeric toration O O Numeric O toration O O Numeric O formation O O O O file Nigh O O O file Numeric O O O	e_Type	infe_Element	X	۲		Modelity	Location	Format	
Interplate Attributes Operator Usives setter Modelity O Iconic O tocation O Numeric O formation O O Usives formation O O Iconic	ight_Plan	Rule Name	Rute3			Seve	Remove	Quit	
Medetity Medetity O Iconic rereft Iconic O formation O rereft O formation O O Iconic O O Insert Clear If If If O If O V O If O If O V Blue THEN Modelity => Ruditory Iormat => Worbal	ight_Plan	م			loerator	,	liatues		
inter Costion formation	wiker	Modelitu				Iconic			
rereft Format Form	ether	tocation		Ħ	0-	Numeric		HL	
rereft O> O< O(= O)= Insert Clear If H = High V O Blue THEN Modelity => Ruditory Formaties Worthal	reruft	Engmat			00	perbat			
O O(= O(= O(= O(= O(= Insert If H = High Y O Blue THEN Modelity => Auditory Format => Upriod	rcraft				O,			I I I	
O (= O)= Insert Clear IF H = High Y o Blue THEN Modelity => Ruditory Format => (JerDal					04				
Insert Clear IF N = Nigh Y <> Blue THEN Modelity => Ruditory Format => Vicront		11		- 11	0.0	1		111	
Open Open Insert Clear If H = Nigh V ← Blue THEN THEN Modelity => Ruditory Format => Upront					0	1		니	
Insert Clear		L		Q	0/~	<u> </u>		KA	
IF N = Nigh Y ← Blue ThEN Modelity → Ruditery Format ⇒ NyrDol		Insert	Clear						
H = High Y © Blue THEN Modelity => Auditory Format => Herbal		11						[범	
Y ⇔ Blue THEN Modelity → Auditory Format ⇒ Herbal		H - High						111	
THEN Modelity -> Auditory Format => Verbal		Y O Blue							
Modelity -> Ruditery Format -> Uerbal		THEN						110	
Format >> Verbal		Modelity ->	Huditery						
NT		Format => (erbal						
								片	
		L						K	

Figure 4(c). Rule3

Figures 4(a)-(c). Rule input dialog

LHS rule patterns are composed of an attribute, an operator, and a value. The list of attributes, displayed in the Attributes list-box, consists of the names of all input and output columns from the first row of the table. Displayed in the Values list-box is the list of value options that correspond to the selected attribute. LHS side patterns are formed by selecting an attribute, an operator (via the Operator radio buttons), and a value. The resulting pattern is displayed at the selected line in the rule display.

RHS rule clauses are composed of an attribute, the assertion operator (=>) and a value. When a RHS clause is being formed, only the names of output columns appear in the Attributes list-box and the Operator radio buttons are not applicable.

New LHS patterns and RHS clauses can be inserted into the rule by pressing the Insert button and deleted by pressing the Clear button. The Save button allows newly created rules to be saved while the Remove button deletes pre-existing ones.

Rule Execution

A single rule can fire multiple times to set attribute values for different information elements. However, rules are applied to the table in a row-wise manner. That is, the LHS and RHS parts of rule can only apply to one row at a time. Figure 5(a) shows the state of table after Rule1 [Figure 4(a)] has executed. Rule1's LHS patterns are satisfied by the values of attributes X and Y for information elements Time_Estimates and Oil_Pressure. Rule1 will fire once for each of these information elements, resulting in their Modality, Location, and Format attributes being set to Visual, Display1, and Iconic, respectively. Rule2 [Figure 4(b)] includes a LHS pattern that matches if "X >= Medium." This is permissible because an order relationship is established on the list of value options. Thus, the relationship (Low < Medium < High) holds for the value options for the attribute X.

Figure 5(b) shows the state of the table after the Rule2 and Rule3 have executed. Notice that because Rule3's RHS does not set the Location attribute, it has not been set for the information elements Altitudes and Fuel_Remaining.

ORIGINAL PAGE IS OF POCR QUALITY

ada_Type inda_Element X Y Maduity Location Format Tight_Plan films_Estimates Low Red Yisual Display I icanic Tight_Plan Altitudes Migh Green Vantaer Wind_Seed Plantum Blue Newther Yisubility High Blue Newther Fuel_Remaining High Red Newther Dil_Pressure Law Red Yisual Display I icanic								-
Plaght_Plagn Time_Estimates Lov Rad Yisual Display I lossic Plaght_Plagn Altitudes High Green Image: State	infe_Type	Info_Element	x	۷	Pladelity	Lecetion	Format	
Flight_Plan Altitudes High Green Image: Constraint of the constra	Flight_Ples	Time_Estimates	Lev	Red	Visual	Display I	lesaic	
Weather Wind_Speed Photlum Blue All Weather Visibility High Blue All Aircraft Fees_Demaining High Blue All Aircraft Fees_Demaining High Blue All Aircraft Dis_Pressure Lew Red Visuel Display 1 Iconic	Flight_Ples	Altitudes	High	Green				
Weether Visibility High Blue Aircraft Fast_Remaining High Rad Aircraft Dil_Pressure Low Rad Visual Display1 Icentc	Weether	Wind_Speed	Medium	51 un				
Mrcruft Fusi_Remaining High Red Mrcruft Oll_Pressure Low Red Visual Display i Icenic	Weether	Visibility	High	81 ue				
Alrcruft Oil_Pressure Lov Red Visuel Display i Icanic	Aircraft	Fuel_Remaining	High	Red				
	Aircraft	OIL_Pressure	Lev	Red	Visuel	Display1	lcenic	



Info_Type Info_Element X Y Modelity Locatten Format Flight_Plan Time_Estimates Low Red Yisual Display I Iconic Flight_Plan Altitudes High Green Auditory Vorbal Vesther Vrind_Speed Fligh Blue Visual Display 2 Numeric Vesther Vrind_Speed Fligh Blue Visual Display 2 Numeric Vesther Vrishillty High Blue Visual Display 2 Numeric Vesther Vrishillty High Blue Visual Display 2 Numeric Vesther Vrishillty High Red Auditory Yerbal Yerbal Ircraft Ott_Pressure Low Red Visual Display 1 iconic				PUI Table			
Tight_Plan Time_Estimates Lov Red Visual Display 1 Iconic Tight_Plan Altitudes High Green Auditory Verbal Vesther Wind_Speed Flodium Blue Visual Display 2 Numeric Vesther Vislalitty High Blue Visual Display 2 Numeric Vesther Vislalitty High Blue Visual Display 2 Numeric Arcraft Fust_Remaining High Red Auditory Verbat Ircraft Oti_Pressure Lov Red Visual Display 1 Iconic	nfo_Type	Info_Element	x	۲	Modelity	Location	Formet
Flight_Plan Altitudes High Green Auditory Verbal Weethor Wind_Space High Blue Visual Display2 Numeric Weathor Visibility High Blue Visual Display2 Numeric Arcruft Fuel_Remaining High Red Auditory Verbal Nrcruft Oll_Pressure Lov Red Visual Display1 toenic	light_Plan	Time_Estimates	Low	Red	Yisuel	Display I	Icentic
Weether Wind_Speed Fladium Blue Visual Display2 Numaric Weether Visibility High Blue Visual Display2 Numaric Mircreft Fuel_Bensining High Blue Visual Display2 Numaric Mircreft Fuel_Bensining High Blue Visual Display2 Numaric Mircreft Fuel_Bensining High Bud Auditory Vorbal Mircreft Oil_Pressure Low Red Visual Display1 Iconic	light_Plan	Altitudes	High	Green	Auditory		Verbel
Weether Visiolity High Blue Visual Display2 Mumaric Aircraft Fuel_Bamaining High Red Auditory Verbal Aircraft OII_Pressure Lov Red Visual Display1 Icentic	Veether	Wing_Speed	filedi use	81w0	Visuel	Display2	Numeric
Alrcroft Fusi_Remaining High Red Auditory Yerbal Alrcraft Oli_Pressure Lew Red Yisual Display1 icentic	Veether	Visibility	High	61 ua	Visual	Display2	Numeric
Alreraft Oli_Pressure Lew Red Yisual Display1 icente	drereft	Fuel_Remaining	High	Red	Auditory		Verbal
· · · · · · · · · · · · · · · · · · ·	ircraft	011_Pressure	Lev	Red	Visuel	Display 1	Icestc
							(

Figure 5(b). Table after Rule2 executes

Figures 5(a) and (b). Effects of rule execution

Once a rule has been entered it will apply to any new information elements that are added to the table. Figure 6 shows the addition of the information element Planned_Speeds in the third row of the table and the setting of its Y attribute to Red. After the Y attribute has been set to Red, Rule3 [Figure 4(c)] will execute and the Modality, and Format attributes will be set accordingly (Figure 7).

The effects of modifying an existing rule are immediately reflected in the state of the table. Figure 8 shows Rule2 modified so that it sets the attribute Location to Display3 instead of Display2. After the modified Rule2 has executed, the Location attribute for information elements Wind_Speed and Visibility will be set to Display3 (Figure 9).

		_		PUI Teble						
nfe_Type	infe_Element	x		٧	Plodelity		Location	Formet		R
light_Plan	Time_Estimates	Lev	_	Red	Visual		Display 1	Icenic		Ľ
light_Ples	Planned_Speeds	High								
Tight_Plan	Altitudee	High F			The second se	- 1		Verbal	7	
veether	Wind_Speed	Thelium	Flig	ght_Plan			Display2	Numeric	-1	7.41
vestior	Vialatitty	High	<u>*</u>		l	_	Display2	Numeric	-11	i.
ferent	Fuel_Remaining	High	L		/[Verbel		
ircraft	OIL_Pressure	Lev	the state	d			Display I	Icenic		
					Ø			· · ·		A STATE OF THE REAL PROPERTY OF THE PARTY OF

Figure 6. Addition of new information element

			1		1	6	
ato_Type	Info_Element	×	۳	riodelity	Location	Formet	
flight_Plan	Time_Estimates	Lev	Red	Yisuel	Display I	Iconic	
Flight_Plan	Planned_Speeds	High `	Red	Auditory	-	Verbel	
flight_Plan	Altitudee	High	Green	Auditory		Verbal	
Westher	Wind_Speed	Madium	61 🗰	Visue:	Display2	Numeric	
Weether	Visibility	High	61 wa	Visuel	Display2	Numeric	
Aircreft	Fuel_Remaining	High	Red	Auditory		Verbet	
Aircroft	OIL_Pressure	Lev	Red	Visual	Otanieul	Icenic	_
		<u> </u>	<u></u>				
				<u></u>]
				<u></u>			

Figure 7. Table after addition of a new information element

Conflicting Rules

A rule conflict is defined as two or more rules attempting to set the same table entry. A conflict situation arises when the PVI designer has entered two or more rules that apply to the same output attribute(s) of an information element. A more subtle conflict situation presents itself when a change in the value of a table entry brings two or more previously non-conflicting rules into conflict.

The tool allows conflicting rules to be entered. However, when multiple rules attempt to set the value of the same table entry, the symbol "CONFLICT" is placed into the entry. The user can then request that a list of all rules that caused the conflict be generated and modify the appropriate rule(s) to resolve the conflict. The resolution of conflicting rules is an important feature of the tool that insures the consistency of the design knowledge.

afe_Type	infe_flement X	٧	Pladality	Location	for met	_
light_Plan	Rule Name Rule2		Save	Remove	Butt	
light_Plan						
iteht_Plan	Attributes	Operato	r	Values	——————— [C	
Aust her	Modality	PI 0-	Display1		MI	
Author	Fermet	00	lisplay 5			
ireraft		0				_
frereft		104				_
		10.	1			-
ĺ					ᆔ	
			<u> </u>		<u> </u>	
					·	
	IF				엄	
	X >= Medium				1 1 1	
	Y - Sive					
	Modelity => Visual					
	Location => Display5					
j.	Format -> Numeric					
					KA	

Figure 8. Modification of Rule2

			PVI Table			
info_Tupo	Info_Element	x	٧	Modelity	Location	Formet
Flight_Plan	Time_Estimates	Lov	Red	Yisuel	DisplayI	Icente
flight_Plan	Planned_Speeds	High	Red	Auditory	<u> </u>	Verbel
Flight_Plan	Altitutes	High	Green	Auditory	· · · · · · · · · · · · · · · · · · ·	Verbei
Wasther	Wind_Speed	Madi um	Sive	Visuel	Display3	Numeric
Weether	Visibility	High	Blue	Visuel	Display3	Numeric
Alreraft	Fuel_Remaining	High	Red	Auditory		Verbal
Aircraft	Oil_Pressure	Low	Ret	Visuel	Display1	iconte
				y		
				s		

Figure 9. Table after modified Rule2 has executed

Column and Value Option Deletion

÷,

The deletion of a column or one of its value options is problematic in that rules that reference the column or value will no longer apply. For example, if the column representing attribute X were deleted from the table, then any rules that reference X will no longer fire. Similarly, if a value option for a column is deleted, any rules referencing that value will no longer execute.

There are several alternatives for handling this situation. One is to remove all rules that make reference to deleted columns or values. This has the dangerous consequence that a large number rules could potentially be removed due to a single operation by the user. A second alternative is to modify the rule set by removing all clauses that make reference to the deleted column or value option. While less severe than the first alternative, a large scale modification of the rule set could occur. A third alternative is to simply alert the user that a number of rules have been invalidated. A query can then be performed over the rule set to locate all invalidated rules. The user then has the option of modifying or deleting these rules as appropriate.

Since Invalidated rules are no longer applicable they are "unfired" by clearing all table entries that were set as a consequence of their execution. When a value option is

414

0-3

removed from a column, all entries in the column with that value are cleared. The clearing of table entries in this manner can result in the LHS patterns of previously executed rules to longer be satisfied. These rules must also be "unfired" to maintain the consistency of the information in the table. The resolution of conflicts, as described above, has not yet been fully implemented. Currently, The truth-maintenance facility of CLIPS Version 5.0 is being considered for use.

IMPLEMENTATION



Figure 10. Tool Architecture

Architecture

The tool is comprised of three main components; a user-interface module, a message handling facility and an embedded CLIPS module (Figure 10). The user interface module is responsible for handling all events generated by the user and operating system. The embedded CLIPS component is responsible for maintaining the table and rule-base. The message handling component facilitates communication between the user-interface and CLIPS modules. Whenever the table is modified by the user, a message is sent to CLIPS by the user-interface. CLIPS handles this message by updating its fact-list and executing any rules that have become activated. If rules that result in the modification of the table are executed, then the CLIPS module sends a message to the user-interface module - which responds by updating the display accordingly. When the user creates rules, the user-interface module sends a message to CLIPS, which responds by adding the rule to its environment using the CLIPS Build function.

Developing With CLIPS On The Macintosh

Programming the Macintosh family of computers requires a substantial investment of time to become familiar with the toolbox routines implemented in ROM. Indispensable to any serious development effort for the Macintosh is the *Inside Macintosh* series (Apple 1985). Knowledge of the basics of Pascal records, procedures and functions is helpful because the toolbox interface is defined in Pascal. Apple's *Programmer's Introduction to the Macintosh Family* (Apple 1988) is a good source of information for those considering a developing for the Macintosh.

Symantec's THINK C compiler was chosen to implement this tool. Since CLIPS for the Macintosh was developed using THINK C, the task of embedding it into an application is made easier by using THINK C. In addition, THINK C comes with a class library that provides an interface to the Macintosh toolbox that can release the programmer from being concerned with many low-level toolbox details. Because Symantec's documentation for the THINK C Class Library (Symantec 1989) is extremely terse and includes few coding examples, Dave Mark's Macintosh C Programming Primer series (Mark and Reed 1989, Mark 1990) is strongly recommended.

CONCLUSIONS AND FUTURE DIRECTIONS

The problem of transferring human knowledge into an expert system is known as the knowledge acquisition bottleneck (Giarratano 1989). This tool helps reduce this bottleneck by facilitating the incremental acquisition of design knowledge from its users. It accomplishes this by allowing users without formal programming experience to easily create rules while assessing their effects. Using CLIPS to represent these rules provides a measure of power that is not attainable using other tools such as a spreadsheet. CLIPS' use of the Rete Algorithm (Forgy 1979) results in the execution of only those rules affected by incremental changes to the table. The pattern matching capabilities of CLIPS allows rules to be formed that reference table entries via user-defined symbols rather than positional row/column addresses imposed by the implementation of the table. This allows rows and columns to be added or rearranged without having to modify any rules.

Future directions for the tool include allowing the formation of more sophisticated rules, incorporating meta-rules that can be used to evaluate and examine the design rules entered by the user, and providing a rule explanation facility.

REFERENCES

Apple Computer, Inc. Inside Macintosh. 6 vols. Addison-Wesley, 1985 - 1991.

- Apple Computer, Inc. Programmer's Introduction to the Macintosh Family. Addison-Wesley, 1988.
- Forgy, Charles L. On the Efficient Implementation of Production Systems. Ph.D. Thesis, Carnegie-Mellon University, 1979.
- Giarratano, Joseph, Gary Riley. Expert Systems Principles and Programming. Boston: PWS-KENT, 1989.
- Mark, Dave, Cartwright Reed. Macintosh C Programming Primer Volume I. Addison-Wesley, 1989.

Mark, Dave. Macintosh C Programming Primer Volume II. Addison-Wesley, 1990.

NASA. CLIPS Reference Manual (Version 5.0). Software Technology Branch, Lyndon B. Johnson Space Center, 1991.

Symantec. THINK C User's Manual. Symantec, 1989.