

INTERACTIVITY AND GRAPHICS AS A SUPPORT FOR THE DESIGN OF CABLE NETS
AND TRUSSES.

MOLLAERT Marijke, ass.prof.
Free University Brussels (V. U. B.)
Department of Architecture
Pleinlaan, 2
BRUSSELS, Belgium

"Quite often the programmer works in a vacuum unaware of the interaction techniques that have been successfully used by other programmers".
M. GREEN, *Computer Graphics*,
January 1983.

"A system has to offer a user friendliness as good as possible in order to be a good tool".
H. HANUSA, *Eurographics '82*.

"A picture makes the data from your computer more understandable, more useful for you. With computer graphics you get the information you need, to make the right decisions faster".
A. PALLER, *Computer Graphics State of the Art Seminar*, 1983.

This paper deals with an important aspect concerning the quality of a computer aided design (CAD) program, namely the human-computer interface. To fix the ideas, the user interface of **FORCE**, a program for the design and visualisation of cable nets and trusses, is considered. The new design of this user interface was necessary to support an easy extension of the dialogue, both alpha-numerical and graphical. The new approach includes a separation of the interaction procedures and the application modules.

INTRODUCTION

By the evaluation of a program, it is important to look at what a user expects from a computer aided design (CAD) program, and what he finds annoying.

The software has to be a reliable support. Being a support can only be achieved if the program is easy to use, and if the output is of the required quality. Reliability means that the program must proceed like the user expects it to proceed. No user likes to notice that he has to restart (an error may never abort the program), that he misunderstood the program, or that he doesn't know what to do next.

CAD-programs need a very good user interface to stimulate the use at full capacity, this implies an understandable man-computer dialogue (in conversational English or graphical), a fast response of the computer, a help facility at any point, the possibility of feed back and a good data editing facility, to process quickly a great number of variations.

A separate user interface management system tries to ensure these requirements.

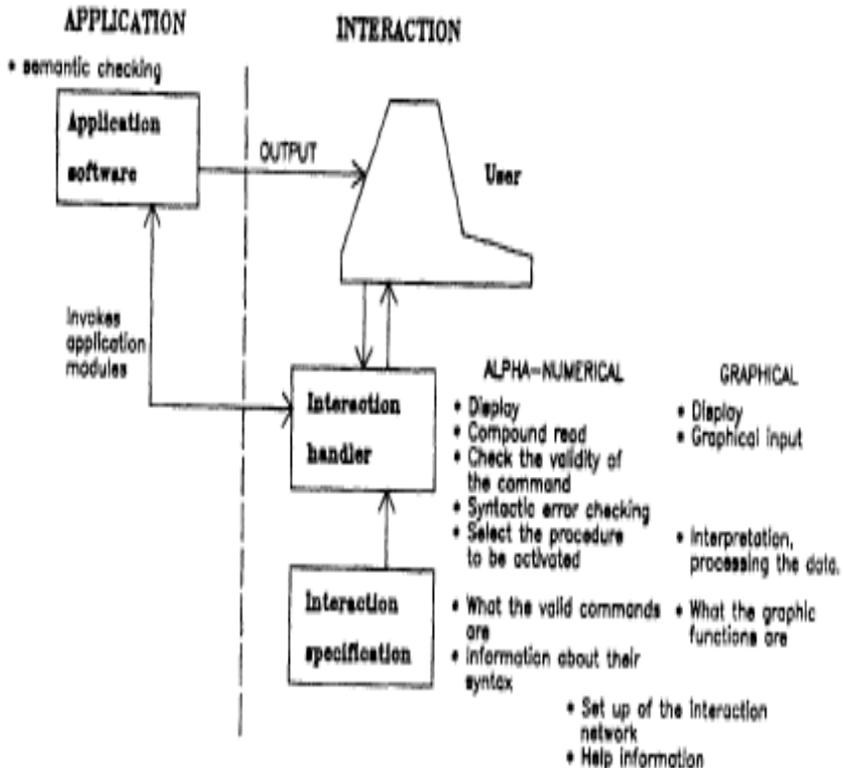
1. THE USER INTERFACE

An interactive command system has been characterized by NEWMAN and SPROULL [1] as a collection of actions and a collection of objects that the actions are to manipulate. Maybe it is worthwhile to add that the actions are activated by intervention of the user. The command system has to respond to whatever the user decides to do next instead of telling the user what to do next.

For the user's acceptance it is necessary to enforce a certain uniformity in the structure of the user interface (for the alpha-numerical display clear the screen and write top down, for the error handling: write the syntactic error messages under the characterstring causing the error, etc.). A certain uniformity corresponds to a certain "style" of interaction. The user interface should increase the usability of the program by clear "help" explanations and error comments.

2. THE USER INTERFACE MANAGEMENT SYSTEM (UIMS), [2]

The management of the user interface can be handled by modules which are separated from the application program, so that there are no user interaction statements left in the application modules. The UIMS can be modelled as follows :



The functions of the UIMS are :

* The definition of the interaction language

This means the specification of the valid graphical functions and the valid commands, with their syntax. The valid interactions are defined by the "interaction-nodes". These nodes are organised in a network. The network can be compared with a railway network, the nodes are the stations, which you can visit or not. Also the way to travel can be chosen by the user. It is the task of the programmer to continue the

the extension and update of the network till he can ensure the efficiency of the program.

* The interaction handler

- An interaction is initiated by a prompt initialising graphical or alpha-numerical mode. The prompt is followed by a read. The input is checked for validity, and the required procedure is activated by the interaction language interpreter.
- The UIMS can contain routines for the exploitation of various physical device characteristics and interaction techniques. (An "interaction technique" is defined as a way of using a physical device to input a certain type of word, coupled with the simplest form of feedback, from the system to the user, [3]). An Interaction Technique Standard is needed to improve the software portability.
- The UIMS can treat the device dependency. Note that even a simple instruction as "clear screen" is different for different terminals, and that there is for example no uniform request to ask the terminal type (without user intervention).
- The UIMS can contain a logical screen handler, to divide the physical view surface into logical portions, so that all logically related information can appear in a specified portion of the screen. This provides an easy understanding. The information consists of alpha-numerical and graphical information. The alpha-numerical data can be subdivided in the display of allowed commands, the response of the user, the help information, the error messages and the result information. Analogously the graphical information can be split up in graphic menu information, a display of the "segments" which can be picked from the database, and multiple views if a 3D object is considered. (It is clear that the logical screens will overlap if the size of the screen is only 80 columns and 20 lines).

3. THE USER INTERFACE OF FORCE. A PROGRAM FOR THE DESIGN AND VISUALISATION OF CABLE NETS AND TRUSSES. BASED ON THE FORCE DENSITY METHOD [4]

3.1. Interaction language definition

The definition of the different interactions is done by the specification and interconnection of the interaction nodes. Each interaction corresponds to a certain task.

TASKTYPE	MEANING	POSSIBLE DEVICE FOR APPLICATION INTERACTION TECHNIQUE
SELECT-ENTITY	pick segment ID	graphic display
POSITION	position WC	tablet, graphic display
SELECT-DECISION	command ID	button, tablet, keyboard
INPUT-SESSION	command ID + data	keyboard
GETCHARACTER	character input	keyboard

The alpha-numerical commands (select-decision, input-session) fulfil a general syntax (APPENDIX 1). Some parameters can be omitted to rely on default values.

The graphical tasks correspond to the graphical functions in GKS.

FUNCTION	MEANING	POSSIBLE DEVICE FOR APPLICATION
* LOCATOR (POSITION)	position WC	graphic display, tablet
VALUATOR	real input	graphic display
CHOICE	integer read	button
* PICK (SELECT-ENTITY)	segment ID	graphic display
STRING	characters	keyboard

Note that there is no unique correspondence between the task-types for the user interface and the functions in GKS, for example the SELECT-DECISION can call for a LOCATOR, a CHOICE, a KEYBOARD (graphic read) or an alphanumeric read, which implies a different technique, although the result for the application program is the same.

3.2. The interaction-node type

The interaction node specifies the type of interaction required, and concentrates for that interaction the data about the allowed commands and

functions.

INTERACTION-NODE= RECORD



ID: STRING20; the name of the node
PROMPT: STRING20; the prompt that will be printed by activation
END: PTRLETTERTYPE; a pointer to a tree of letters, giving for each command the allowed strings
USE WHAT-KIND: TASKTYPE OF
SELECT-DECISION: (DECISIONS: PTCOMDATA); pointer to the allowed decision records
INPUT-SESSION: (COMMANDS: PTCOMDATA); pointer to the allowed command records
GETCHARACTER: (ONE-CHAR: PCHAR-REC); pointer to a character record
SELECT-ENTITY: (PICKER: PENTITY-REC); pointer to a list of entity records
POST: (LOCATE: PPOST-REC); pointer to a position record

END;

COMDATA=RECORD



PARAMSTR: STRING20; symbolic parameterizing
INSTNAME: STRING20; the command name as a string
COM. COMTYPE: the command identifier (lower type)
DESCRIPTION: THREE-LINES; help information
NEXT-TO-DO: PINTERACTION-NODE; pointer to the next interaction-node
ALTERNATIVE: PTCOMDATA-REC; pointer to an alternative command

END;

CLASS=RECORD



NEXT-TO-DO: PINTERACTION-NODE; a pointer to the next interaction-node
CHARAC: CHAR; the pushed key

END;

ENTITY=RECORD



ID: INTTYPE; identifier of the picked segment
BUTTON: INTTYPE; identifier of the pressed button
NAME: STRING10; name of the segment
NEXT-TO-DO: PINTERACTION-NODE; a pointer to the next interaction-node
ALTERNATIVE: PENTITY-REC; a pointer to another valid segment (graphic function)

END;

POINT=RECORD

BUTTON: INTTYPE; identifier of the pressed button
X: REAL; x, y coordinates of the chosen point
Y: REAL; "
NEXT-TO-DO: PINTERACTION-NODE; a pointer to the next interaction-node

END;

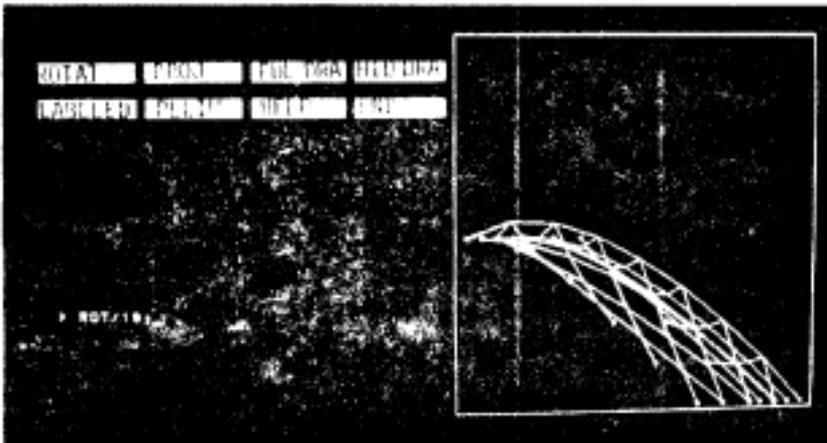
* This data is the result of the interaction; alphanumeric information is buffered in a list of 'tokens', other data is kept in the interact on-node

3.3. The interaction network

The pointerfields included, allow the construction of the network. The total network of the interaction nodes of FORCE is given in APPENDIX 2. As an example we can consider the HELP facility.

Assume that the user arrived at node DRAWPART (1), and has on the display a menu with 6 commands (DRAWP, LIST, HELP, LLU, PASS, END). He can push the HELP function key to ask for more information (no syntactical errors are possible). The program goes to the next interaction node (2). The screen is cleared and the requested information listed. The user can read this information. A "press return to continue" is prompted after the HELP information. The user pushes the return button to continue, and the program returns to the previous interaction node (3).

- for the input : a read routine for alpha-numerical input, a pick routine to pick a graphical segment, and a locator to enter X,Y coordinates
- if alpha-numerical input was given : a test for the validity of the command, and a test for the validity of the parameterstring.
- an interpreter routine to activate the right procedure
- an error routine to display the error message
- a help routine.



The screen is viewed as an alpha-numerical terminal as long as no graphic task is started. Once this is done, two logical screens are generated, the second allowing the graphic intervention (or output). The function keys allow an easy choice of the successive modules to be activated.

3.5. Graphical editing

Visualisation is very important to evaluate a design. Specially for architectural designs, it is necessary to move around and inside the volumes. Graphical editing allows to manipulate the geometry of the objects and their organisation.

As an example, we can copy and move a "segment" (a graphical object on the screen). As soon as the COPY instruction is initiated, a first interaction asks for an entity to be picked, once this is done the next interaction is started automatically, and "reads" the position where the entity has to be copied.

Note that there is no possibility for a syntactic error in this sequence neither the user has to remember "what to do next?", which implies a natural and easy dialogue.

Sometimes computer graphics can give you results that you cannot get from any other technique.

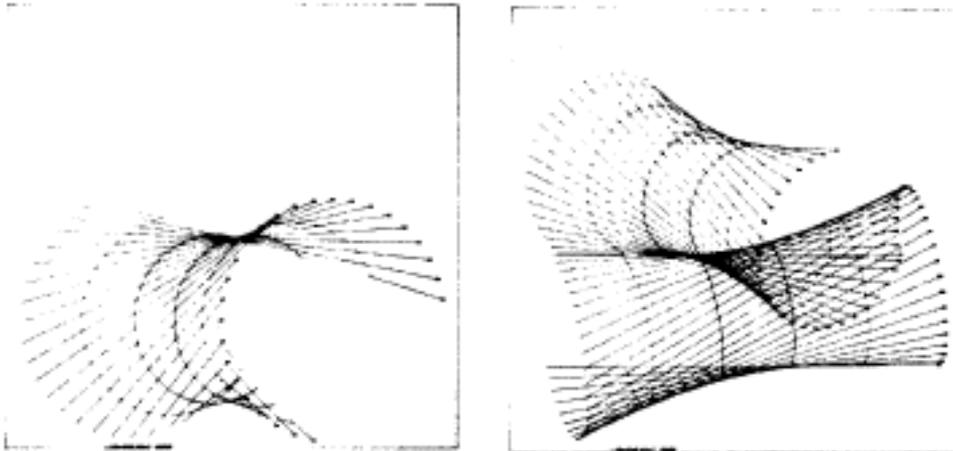
Consider the following. FORCE computes the form of tension structures. Hyperboloides are such forms. Besides they are ruled surfaces. They can be cut in pieces along their generators. A global surface can be generated by connecting different pieces of hyperboloides in a tangential way. These connections can be handled by the program.

Pieces of a hyperboloid can be copied. For two pieces the user picks two generators and three points, necessary to do the calculations for tangential connection, and the two hyperboloids brought together are drawn on the screen.

Once this is done the user can look at the result from different positions.

A lot of assemblies can be generated and compared with each other.

For the computer there is no difference in treating curved surfaces or regular boxes. For human beings, the visualisation and editing of curved surfaces is a larger support.



3.6. Graphical output preparation

It is important to allow the user to specify the output he wants: the format of the plots, how many plots on one sheet, text information on the plot and so on. The program provides drawings, but gives also the possibility to make diagrams of the numerical data.

4. FINAL REMARKS

From the previous can be concluded that the user interface is more than a question/answer system. A systematic design, with clear visualisation, is important. To include graphics in the interactivity is just a step nearer to the natural way of designing.

The author agrees with the opinion, that designing an interface is a multidisciplinary task. A second fact is the lack of standardisation in the instructions of a terminal, which makes the design of more powerful portable interfaces almost impossible.

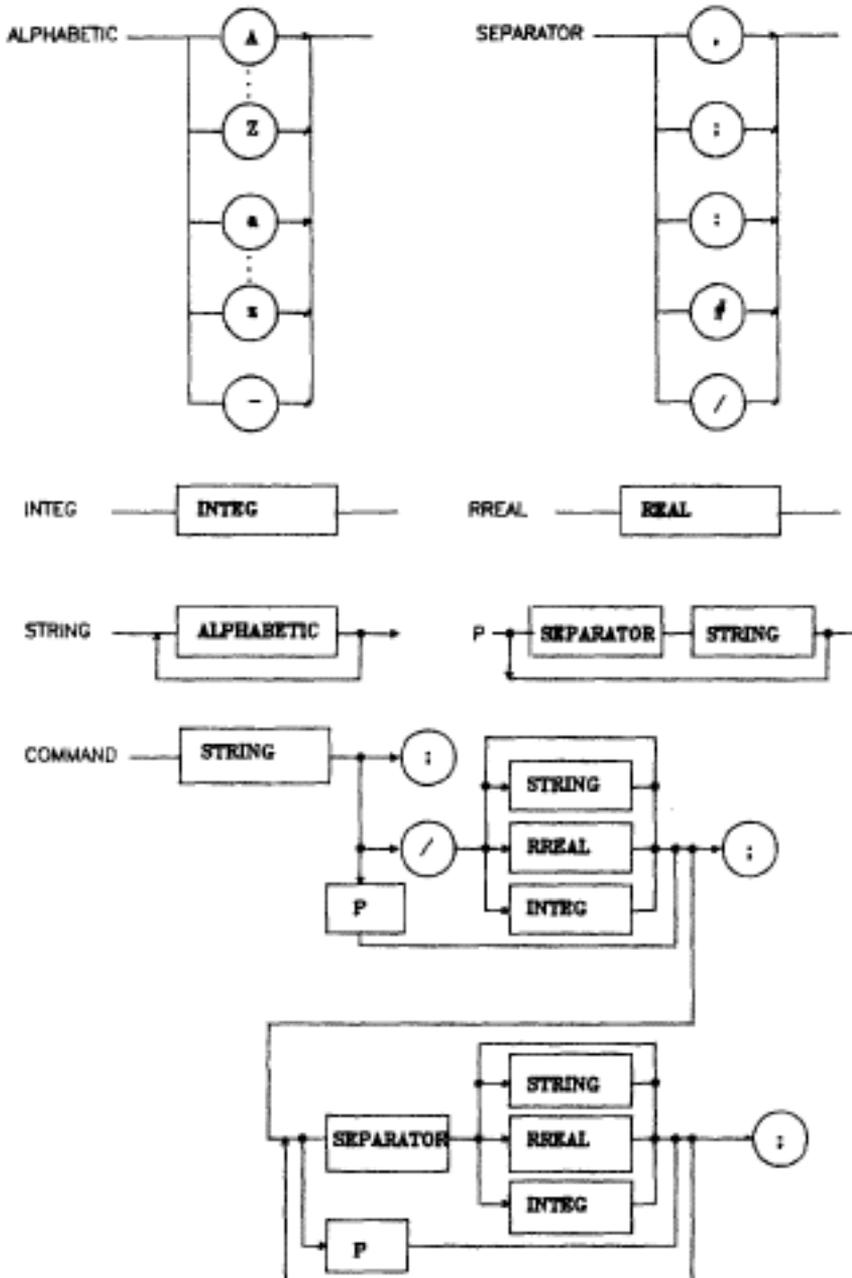
Maybe that if the problem becomes known that some work will be done on the standardisation and the development of interaction packages.

REFERENCES

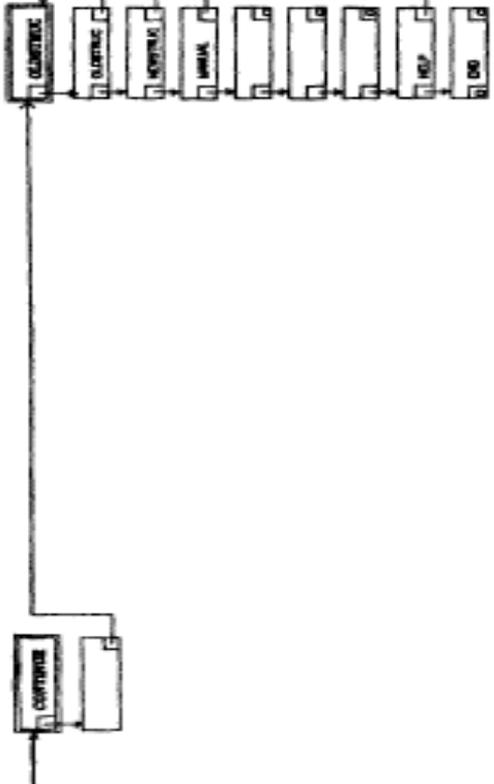
- [1] NEWMANN, W.M. and SPROULL. F.R., Principles of Interactive Graphics. New York, McGraw-Hill, 1979.
- [2] JAMES, J.T. and GRIFFITH, H.. Graphical Input Interaction Technique. Computer Graphics, 17.1, January 1983.
- [3] FOLEY, J.D., WALLACE, Y.L. and CHAN, P.M., The Human Factors of Interaction Techniques.
IIST Report 81-03, Department of Electrical Engineering and Computer Science, The George Washington University, Washington.
- [4] SCHEK, H.J., The Force Density Method for Form-finding and Computation of general Networks.
Computer Methods in Applied Mechanics and Engineering 3 (1974), 115-134 pp.
- [5] KAMRAN, A. and FELDMAN, M.B., Graphics Programming Independent of Interaction Techniques and Styles.
Computer Graphics, 17.1, January 1983.

APPENDIX 1

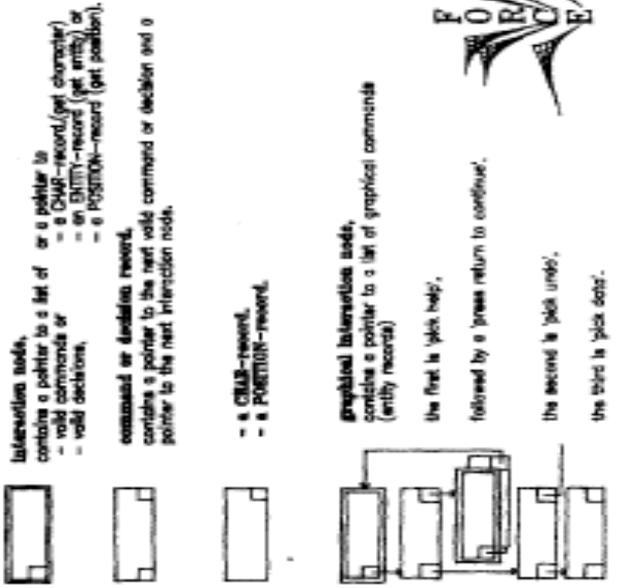
Syntax of the interaction language:



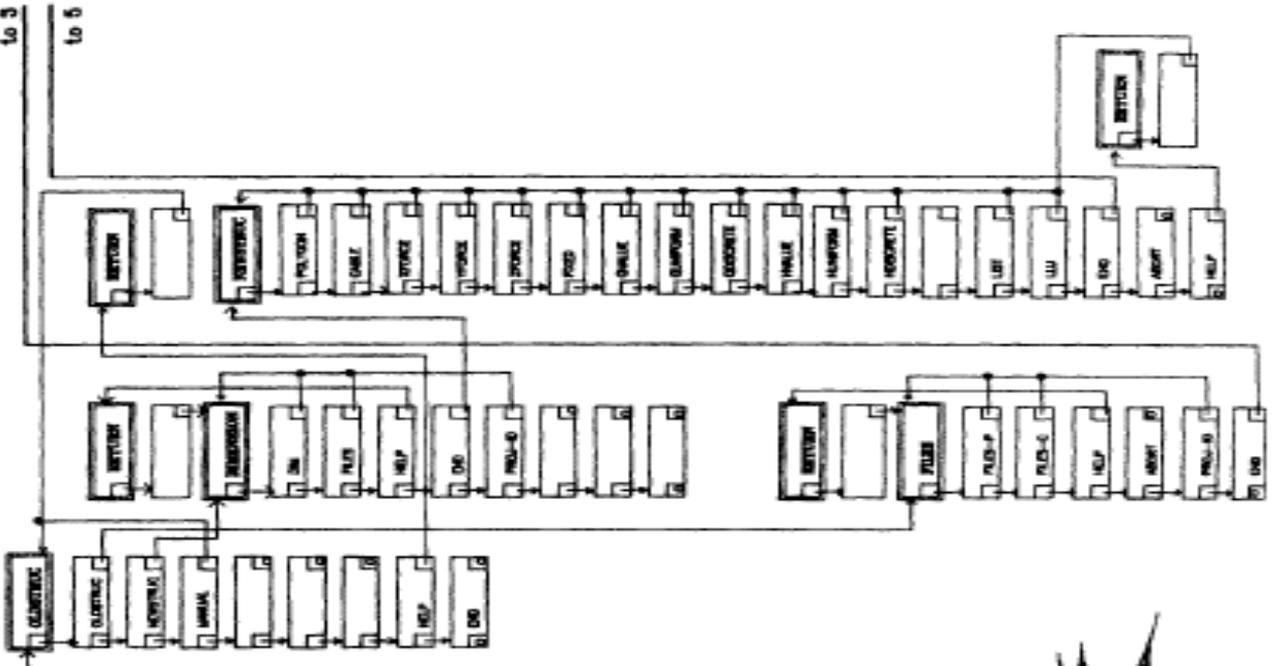
The interaction network of FORCE. 1.



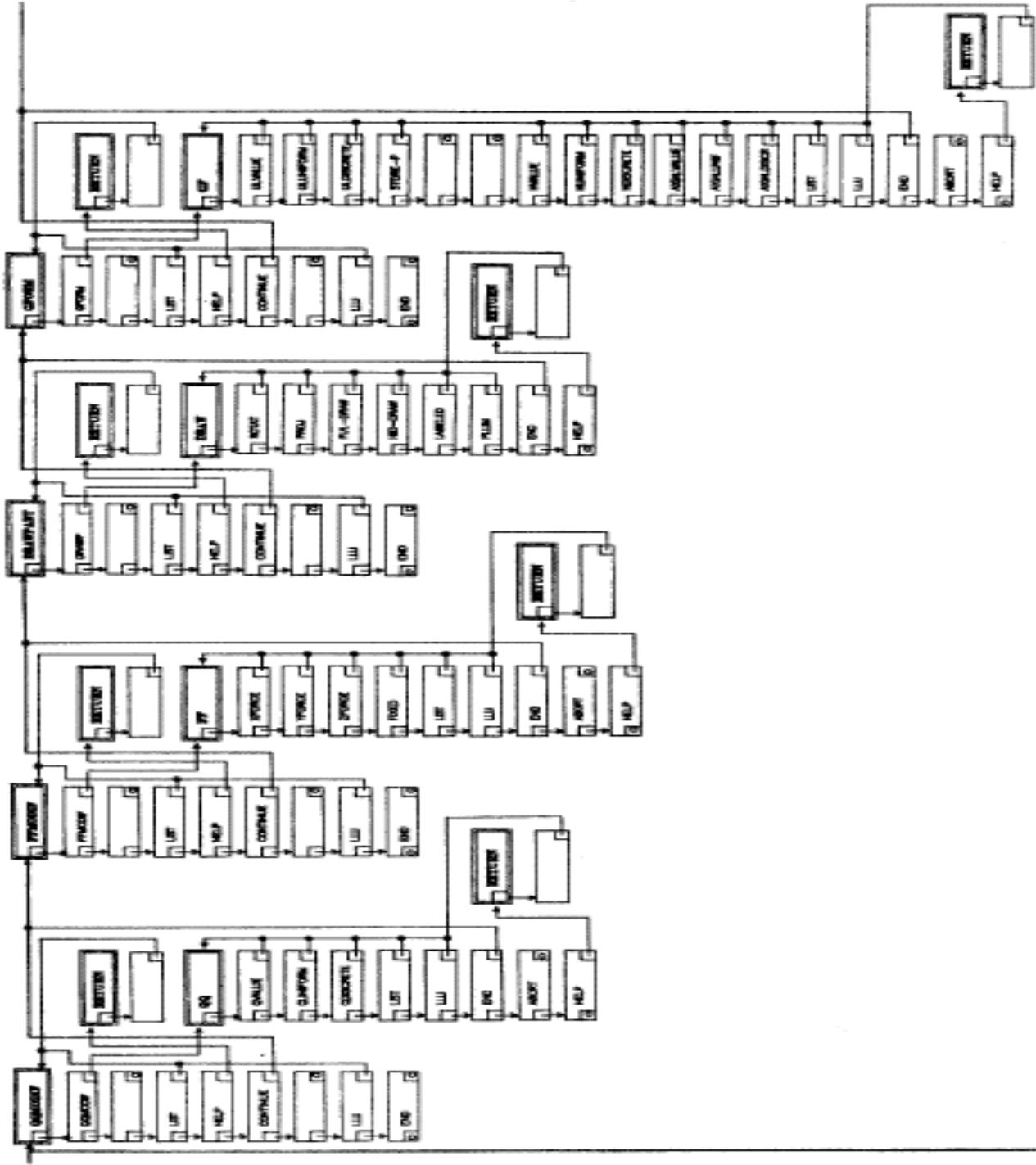
APPENDIX 2



The interaction network of FORCE. 2.



The interaction network of FORCE. 3. The interaction network of FORCE. 4. The interaction network of FORCE. 5. The interaction network of FORCE. 6.



**Order a complete set of
eCAADe Proceedings (1983 - 2000)
on CD-Rom!**

**Further information:
<http://www.ecaade.org>**