PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :		(11) International Publication Number:
G06F 13/00, 15/16	A1	WO 93/21584
		(43) International Publication Date: 28 October 1993 (28.10.93)

(21) International Application Number: PCT/US93/03258

(22) International Filing Date: 7 April 1993 (07.04.93)

(30) Priority data:
07/865,107 8 April 1992 (08.04.92) US

(71) Applicant: CEGAB DEVELOPMENT CORPORATION [US/US]; 7052 W. 64th Street, Chicago, IL 60638 (US).

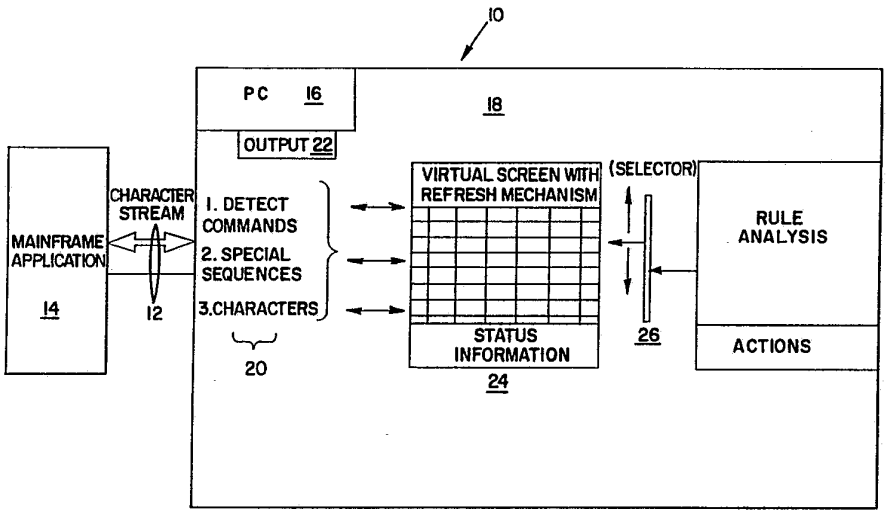
(72) Inventor: KMIEC, Slawomir, A. ; 306 The West Mall, Apt. 307, Etobicoke, Ontario M9C 1C8 (CA).

(74) Agent: RETSKY, Jonathan, E.; Willian Brinks Olds Hofer Gilson & Lione, 455 North Cityfront Plaza Drive, Suite 3600, Chicago, IL 60611-5599 (US).

(81) Designated States: AT, AU, BB, BG, BR, CH, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: INTERFACE SYSTEM FOR COUPLING TWO COMPUTER ENVIRONMENTS



(57) Abstract

A computer interface (10) is provided for communication between a first computer environment (14) and a second computer environment (16). The interface (10) includes an input (50) for receiving information from the first computer environment (14) and an output for communicating information to the second computer environment (16). Information received from the first computer environment (14) is stored in a memory for analysis. A plurality of predetermined rules are provided for translating information stored in the memory to a form compatible with the second computer environment (16). A loop routine is employed for sequentially testing whether a specific rule from the predetermined set of rules should be implemented. The loop routine includes a mechanism for identifying a subset of rules, from the predetermined set, which are to be executed in response to the received information. Each rule has associated with it an area of sensitivity, and a mechanism is provided to account for overlapping areas of sensitivity for different rules. The area of sensitivity for a specific rule is flagged when information received from the first computer environment (14) impacts upon that area of sensitivity.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SK	Slovak Republic
CI	Côte d'Ivoire	KZ	Kazakhstan	SN	Senegal
CM	Cameroon	LI	Liechtenstein	SU	Soviet Union
CS	Czechoslovakia	LK	Sri Lanka	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	UA	Ukraine
DK	Denmark	MG	Madagascar	US	United States of America
ES	Spain	ML	Mali	VN	Viet Nam
FI	Finland	MN	Mongolia		

INTERFACE SYSTEM FOR COUPLING TWO COMPUTER ENVIRONMENTS

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by any one of the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

FIELD OF THE INVENTION

The present invention relates generally to an interface between a first computer environment and a second computer environment, and more specifically, to a system for logically translating information received from a first computer for use on a second computer.

BACKGROUND OF THE INVENTION

Since most mainframe computers are text-oriented, mainframe computer operators have communicated information (i.e. data and commands) to and from the mainframe computer through relatively unsophisticated cathode ray tube ("CRT") terminals. Such CRT terminals are connected either directly to the mainframe computer, or are connected remotely through a telephonic or other data transmission link. Through these "dumb" terminals, operators commonly send commands and other arguments to the mainframe through a command line displayed on the terminal. In most mainframe computer environments, more than one of these unsophisticated terminals can be connected for communication to the mainframe.

This form of communication between a mainframe computer and its operators suffers from a number of disadvantages. Primarily, due to the limited capabilities of the terminal (i.e. low resolution) and slow rate of information transmission, information received from the mainframe and displayed on the dumb terminal often appears in a cryptic format. Although certain software packages, such as Microsoft Corporation's Windows 3.0, include some rudimentary graphics, only characters and numerics are communicated to the program work area while the graphics appear as periphery.

The mainframe computer also dictates the format and type of information communicated to the operator. Thus, this type of user interface is not very "user friendly". Nor, due to the lack of software and hardware provided on the terminal, does the user have the option of customizing the user interface (i.e. the appearance of the display screen) to a visually or otherwise desirable format.

Moreover, in such a computer system, each mainframe computer and its associated CRT terminals are stand-alone entities. As such, the user interface of one mainframe computer can differ markedly from that of another mainframe computer. This results in a multiplicity of differing computer environments, each for a separate mainframe computer. As a consequence, a computer operator must learn multiple user interfaces in order to be proficient on various mainframe computers running different software packages. What is lacking is a generic or universal interface that can be used with many different mainframe computers.

A universal interface would allow an operator to communicate with a host of mainframe computers, each running separate programs, without having to use or learn a separate user interface for each computer. Additionally, a universal user interface will allow the operator to

customize the graphical appearance of data received at his or her display.

With a universal computer interface, the prior dumb CRT terminals can be replaced by personal computers running interface software or using a hardware interface card. By employing a personal computer, the operator can benefit from the software and hardware included in such a device to make the user interface more user friendly. In addition, the operator can combine communication to and from a mainframe computer with other hardware or software applications accessible on the personal computer. Such a universal interface would also increase the power and performance of the personal computer by allowing easy access to a remote mainframe computer.

SUMMARY OF THE INVENTION

In view of the above, a computer interface is provided, which has an input device capable of receiving information from a first computer environment. Information received from the first computer environment is stored in a memory to be accessed by the interface. Included in the interface are a plurality of predetermined rules. Each rule is employed to logically translate information stored in the memory to a format compatible with a second computer environment. The interface includes a program containing a loop routine for selectively testing each of the predetermined rules, in response to the received information, to determine if any rule should be implemented. An output is provided for communicating the translated information to the second computer environment.

In a preferred embodiment of the invention, the loop routine is cycled to test for any rules necessary to be implemented and initiate the subroutines accompanying selected rules. In this embodiment, information received by the interface is stored in memory in the form of a virtual screen in the manner in which the information would

be displayed on a CRT terminal. In the preferred embodiment, the first computer environment comprises a mainframe computer and the second computer environment comprises a personal computer.

In another aspect of the invention, a flagging routine is provided to optimize the loop routine. According to the invention, the flagging routine is provided for selecting a subset of rules from the predetermined set of rules which are to be implemented in response to the received information. In a preferred embodiment of the invention, the interface tests predetermined areas of sensitivity associated with each of the predetermined rules. Information received from the first computer environment is analyzed for its relation to any of the predetermined areas of sensitivity. If the information received requires any one of these rules to be implemented, its associated area of sensitivity is marked through the flagging routine. In the preferred embodiment, a count routine is also provided to account for overlapping areas of sensitivity for different rules.

These and other features and advantages of the invention will be further understood upon consideration of the following detailed description of the presently preferred embodiments of the invention, taken in conjunction with the appended drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of one presently preferred embodiment of the computer interface of the invention;

FIG. 2 is a flowchart of the routines performed by the computer interface of FIG. 1, where FIG. 2(a) shows the initialization of the interface and the loop routine, and FIG. 2(b) shows the communication routines performed at the front end of the interface;

FIG. 3 is a hardware representation of the preferred computer interface of the invention;

FIG. 4 illustrates the manner in which incoming data is stored and flagged in the preferred interface shown in FIG. 3, where FIG. 4(a) shows the correspondence between the virtual screen and flag screen, and FIG. 4(b) illustrates the areas of sensitivity associated with the various rules of the invention;

FIG. 5 illustrates the correspondence between the rules, the flag screen and the count screen of the preferred embodiment of the invention;

FIG. 6 illustrates a mechanism for accounting for the overlapping areas of sensitivity shown in FIG. 5; and

FIG. 7 graphically illustrates the flow of data in the preferred computer interface of the invention and the loop routine for scanning the rules.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to FIG. 1, what is shown is a block diagram of one presently preferred embodiment of the computer interface of the invention, generally designated at 10. As shown in FIG. 1, an input/output ("I/O") link 12 is provided to connect a first computer 14 with a second computer 16. In most embodiments, the operating environment of the first computer 14 is not compatible with that of the second computer 16. Alternatively, the graphical user interface of the second computer 16 may advantageously differ from the user interface of the first computer 14. Interposed between the remote computer 14 and the second computer 16, therefore, is the computer interface 18 of the invention.

As shown in FIG. 1, the computer interface 18 includes an input 20 to receive information from the first computer 14, and an output 22 to communicate translated information to the second computer 16 for display. In the preferred embodiment of FIG. 1, the first computer 14

comprises a mainframe computer and the second computer 16 comprises a personal computer or the like.

As those skilled in the art will appreciate, the computer interface 18 of the invention can be configured in many different computer environments without departing from the spirit and scope of the invention. For example, the second computer 16 can comprise a smart CRT terminal including resident hardware and software. The computer interface 18 can be additionally configured to facilitate communication between two mainframe computers, or to facilitate communication over a computer network such as a local area network. The computer interface 18 can, of course, be configured to communicate with another computer interface 18, or even between differing logical computer environments.

Included in FIG. 1, is a visual representation of the primary features of the computer interface 18. Commands and characters received at the input 20 of the computer interface 18 are communicated to, and stored in, a memory 24. As will be discussed in greater detail below, the commands and characters received are stored in a manner representative of their appearance on a display terminal coupled to the first computer 14. In the computer interface 18, the commands and characters received from the first computer 14 are not immediately displayed on any display device connected to the interface 18. Instead, the commands and characters stored in the memory 24 are first translated to a format compatible with the second computer 16. The ultimate format of the displayed information is dictated either by user preference or the requirements of the second computer environment.

Information stored in the memory 24 is thus stored in the form of a "virtual" screen. With a virtual screen, the computer interface 18 tests and analyzes the stored commands and characters, and translates this information into a format compatible with the second

information is appropriately translated, it is communicated to the second computer 16 (i.e. to be displayed to the operator).

A visual representation of the "rule" analysis program employed with the computer interface 18 is also shown in FIG. 1. As illustrated, a selecting function is provided to selectively analyze the commands and characters appearing on the virtual screen stored in the memory 24. This selecting function comprises a loop routine in the preferred embodiment of the invention, and is graphically represented by the selector 26 shown in FIG. 1.

According to the loop routine, each location of the virtual screen stored in the memory 24 is analyzed to determine whether any information "displayed" in that section of the virtual screen needs to be translated. As further defined below, the determination of whether to translate any information stored in the memory 24 involves the execution of various rules preprogrammed into the computer interface 18. Implementing a rule involves both checking for the rule's applicability, and if applicable, performing the actions associated with that rule to translate the stored information for compatibility with the second computer environment.

In the preferred embodiment of the computer interface 18, a rule is defined as a logical unit of operation and any given rule may operate on information received and stored in the memory 24 in the form of the virtual screen. Conversely, each unit of information stored in the memory 24 can affect or implicate one or more rules provided in the computer interface 18. Thus, the computer interface 18 employs the predetermined set of rules to logically translate information received from one computer environment to a format compatible with a second computer environment.

As those skilled in the art will appreciate, a specific set of rules is not provided with the computer

interface 18. Since the computer interface 18 is generic and universal to all computer environments, a specific set of rules, which are tailored to a specific application of the interface, must be developed for each such application of the interface. Each predetermined set of rules is thus developed by the operator or programmer of a specific computer to tailor the computer interface 18 to that specific computer environment. Accordingly, a tool kit is preferably provided with the computer interface 18 to allow programmers or operators to develop such rules for the particular computers connected to the computer interface 18.

Although defined as a logical unit of operation, a rule itself can involve many subtasks. In fact, a rule may initiate a subroutine to perform whatever operations are necessary to logically and physically translate the received information into a format compatible with the second computer environment. Thus, each rule is tailored to the particular needs of the computer environment and application programs to which the computer interface 18 is attached.

The computer interface 18, therefore, includes a facility to receive information communicated from the first computer 14. Information received at the input 20 is preferably coupled to a terminal emulation program included in the computer interface 18. Information received by the computer interface 18 is first directed to the terminal emulation program. The terminal emulator converts the received information to a format compatible with the interface, and then redirects the received information to the virtual screen stored in memory 24. As already mentioned, the information received by the terminal emulator is not displayed on any actual display screen in its received format.

The computer interface 18 next checks the predetermined set of rules to determine if and how to

respond to the received information. The rules themselves perform the logical translation necessary to communicate the received information to the second computer environment. As described below, an optimization technique is employed in a preferred embodiment of the interface so that all rules are not always checked for every sequence of received information. Finally, the information translated by the respective rules is communicated to the second computer 16.

The computer interface 18 preferably also includes modules for hardware support and system optimization. In one alternate embodiment, hardware support modules are provided to receive input from either a mouse or an audio source. Thus, such modules facilitate support for devices not readily available on the first computer 14.

The preferred embodiment of the computer interface 18 also includes optimization techniques to reduce the amount of signal traffic over the I/O link 12. By limiting such traffic to and from the first computer 14 to essential data, the transaction bandwidth over the I/O link 12 is reduced to a minimum, thereby either reducing costs or allowing for additional applications (i.e. via additional rules) to be implemented on the computer interface 18, or speeding up the appearance of the first computer 14.

Referring to FIG. 2, a flow chart of the routines performed in one presently preferred embodiment of the computer interface 18 is provided. As shown in FIG. 2(a), after start-up of the computer interface 18, the proper environment of the interface is set at step 30. This environment primarily includes initializing the terminal emulator for communication to the particular first computer environment. Connections are established to the first computer 14 and a log-in process is initiated at step 32. After the computer interface 18 has been initialized, the

interface enters the loop routine at step 34. As mentioned above, the loop routine is provided in the interface to analyze which rules need to be implemented in response to information received from the first computer 14. Each rule is associated with an "action" 36, where each action performs the logical translation necessary to convert the information stored in the memory 24 to a format compatible with the second computer environment.

The flow chart shown in FIG. 2(b) illustrates the communication tasks performed at the front end of the computer interface 18. At step 40, characters are received at the input 20 coupled to the first computer 14. These character sequences are analyzed at step 42 and ultimately communicated through the terminal emulator described above to the memory 24. At step 46, a response is provided to the first computer 14 as a result of rule implementation, or the result of incorporating other sources of information (i.e. from a mouse or audio input) to be communicated to the first computer 14. Finally, a log-off routine is provided at step 48 and the proper environment of the interface is restored upon exiting communication with the first computer 14.

A hardware representation of a preferred embodiment of the computer interface 18 is provided in FIG. 3. Information received from the first computer 14 is communicated over the I/O link 12 to a programmable terminal driver or terminal emulator 50. The terminal emulator 50 is employed to configure a personal computer to appear as a specific dumb terminal to the first computer 14 in a manner generally known in the art. Information received by the terminal emulator 50 is then stored in the form of a virtual screen 52 in a section of readable/writable memory. As described below, data is then transferred to a second memory-resident screen, or "flag" screen 54, which is employed to optimize the performance of

the loop routine by reducing the subset of rules required to translate the received sequence of characters.

The flag screen 54 is employed in connection with a flagging routine to reduce the necessity for checking each and every rule in the predetermined set of rules in response to each sequence of characters received from the first computer 14. The loop routine scans the virtual screen 52 to determine if any rule needs to be implemented. Each rule, as it is checked by the loop routine, is preferably associated with a specific portion of the virtual screen 52. The flagging routine reduces the number of rules the loop routine must check by identifying only the various portions of the virtual screen 52 impacted by the received character sequence.

Referring to FIG. 4(a), incoming data 60 received from the terminal emulator 50 is stored in its appropriate location on the virtual screen 52. In response to the incoming data 60, a "flag" 62 is set on the flag screen 54 in an area corresponding to the placement of the incoming data 60 on the virtual screen 52. The flag screen 54 thus comprises a shadow screen for the virtual screen 52, and all the flags 62 of the flag screen 54 are initially cleared on reset of the computer interface 18. Since only certain areas of a display screen are affected at any given time by the incoming data 60, the flagging routine optimizes the loop routine to check only those areas currently implicated.

As shown in FIG. 4(b), each rule 64 is preferably associated with a specific area of sensitivity 66 on the flag screen 54. Constraining each rule to a specific area of sensitivity 66 is practicable since information displayed on a CRT terminal is usually associated with specific areas of the terminal screen in light of the function or purpose of that information. Since each rule 64 is preferably responsive to information that would be displayed in a specific area on a display screen, the

flagging routine greatly increases the efficiency of the computer interface 18. By employing the flagging routine, only rules 64 associated with flagged areas of sensitivity 66 need be checked, rather than scanning the entire virtual screen 52 and implementing all of the rules 64. The areas of sensitivity 66 are thus preprogrammed along with the development of each rule 64, and each area of sensitivity 66 can take any shape or configuration desired.

Since areas of sensitivity 66 can have such random shapes, there exists the possibility that some areas of sensitivity 66 may overlap. The flagging routine takes into account the overlapping nature of the areas of sensitivity 66 and allows for prioritization of areas of sensitivity 66 that are to be analyzed first. Other methods for optimizing the loop routine can be employed, but are less preferred than the flagging routine. In one alternate embodiment, an area of the display screen can be copied to memory and compared periodically to determine whether it has changed. In a second alternate embodiment, a flag can be assigned to each pixel of a display screen, but such a solution would not allow for overlapping areas of sensitivity 66.

In order to facilitate overlapping areas of sensitivity 72, the flagging routine is implemented with yet another virtual-type screen stored in memory. Referring to FIG. 5, a count screen 70, having a one-to-one correspondence with the flag screen 54, and thus the virtual screen 52, is included in the computer interface 18. The count screen 70 is employed to account for overlapping areas of sensitivity 72 and includes flags 68 of its own, as described below.

According to the flagging routine, as incoming data 60 is stored in the virtual screen 52, flags 62 associated with those locations are set in the flag screen 54. By defining each count screen flag 68 as an integer, the flags 62 of the flag screen 54 can also be set to

integer values to indicate the number of dependent rules, as shown in FIG. 5. The flags 62 on the flag screen 54 are thus set to values from corresponding flags 68 appearing on the count screen 70. On reset of the computer interface 18, these count screen flags 68 are also cleared to all zero values (FIG. 6). For overlapping areas of sensitivity 72, therefore, the integer value of each count screen flag 68 reflects the number of rules associated with that specific area of sensitivity 66 affected by the incoming sequence of characters.

The flagging routine thus cooperates with the loop routine to check only those rules whose associated areas of sensitivity 66 were touched by the incoming data 60. Unless the flags 62 appearing on the flag screen 54 for a specific rule 64 are greater than zero, that rule 64 need not be implemented. Conversely, a specific rule 64 is determined to be implemented by detecting a non-zero value in its area of sensitivity 66 appearing on the flag screen 54. After implementation of each of these rules 64, the flags 62 appearing in that rule's area of sensitivity 66 are decremented by one. The loop routine thus proceeds implementing rules until all of the flags 62 on the flag screen 54 are returned to their zero values.

The flagging routine thus reduces the number of rules 64 performed per cycle of the loop routine to those rules 64 associated with regions of the virtual screen 52 that have changed in response to incoming data 60. The flagging routine increases the speed and efficiency of the computer interface 18 by reducing the mean time between rule implementation. Without such speed and efficiency, the computer interface 18 may otherwise be too slow to perform, or even allow for, certain programs executing on the first computer 14 (FIG. 1).

According to one preferred embodiment of the invention shown in FIG. 7, the loop routine executes in a cyclical fashion generally returning to Rule No. 1 upon

completion of each cycle of the loop. In another preferred embodiment, the looping routine can provide for branching or jumping to critical rules. However, even in this alternate embodiment, all rules 64 identified by the flagging routine are performed before completing the loop. Thus, when programming rules 64 for use with the computer interface 18 it may be advantageous to place any important or system critical rules toward the beginning of the predetermined set of rules. A reset feature is also provided in an alternate embodiment where, upon reset, each and every rule is performed and the entire display screen is re-painted. A copy of a preferred program for use in the computer interface 18, including the loop routine and flagging routine, is provided in the appendix.

The computer interface 18 thus provides a fast and efficient system for communication between one computer environment, such as a mainframe computer, and another computer environment, such as a programmable computer, which are otherwise incompatible. In such a system, specific rules are developed for logical translation of information received from the first computer environment to a format compatible with the second computer environment. The rules themselves are developed by the programmer or operator, but execute through a loop routine included in the computer interface 18. A flagging routine is also provided to increase the efficiency of the computer interface 18 by performing only those rules affected by recent data transferred from the first computer environment.

The invention may be embodied in other forms than those specifically disclosed herein without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive, and the scope of the invention is commensurate with the appended claims rather than the foregoing description.

I CLAIM:

1. A computer interface comprising:

an input for receiving information from a first computer environment;

first means for storing said received information;

means for checking a predetermined set of rules, said rules for selectively logically translating the stored information to information for use in a second computer environment; and

an output for communicating a signal representative of the translated information.

2. The computer interface defined in Claim 1, wherein the means for checking the predetermined set of rules comprises a loop routine that tests each rule in a predetermined sequence to ascertain if said rule should be implemented.

3. The computer interface defined in Claim 2, wherein the means for checking further comprises means for retrieving from said means for storing information associated with a first rule if said first rule is determined to be implemented.

4. The computer interface defined in Claim 1, wherein each rule comprises a subroutine of operations necessary to translate said stored information.

5. The computer interface defined in Claim 1, wherein said means for storing stores said received information in a form representative of the manner said received information appears on a display screen.

6. The computer interface defined in Claim 1, wherein said signal representative of the translated information is communicated to a personal computer.

7. The computer interface defined in Claim 1, wherein said signal representative of the translated information is communicated to said second computer environment.

8. The computer interface defined in Claim 1, wherein said input is coupled to a mainframe computer.

9. The computer interface defined in Claim 1, wherein said input is operative to receive information from a mouse.

10. The computer interface defined in Claim 1, wherein said input is operative to receive information from an audio source.

11. The computer interface defined in Claim 1, wherein said input is coupled to a local area network.

12. The computer interface defined in Claim 1, further comprising means for identifying a subset of rules from the predetermined set of rules to be implemented in response to said received information, wherein said means for identifying tests predetermined areas of sensitivity associated with said subset of rules.

13. The computer interface defined in Claim 12, wherein said means for identifying a subset of rules further comprises means for flagging rules determined to be implemented from the predetermined set of rules.

14. The computer interface defined in Claim 13, further comprising second means for storing a representation of the areas of sensitivity of said flagged rules.

15. The computer interface defined in Claim 14, wherein a signal indicative of received information associated with a first area of sensitivity is stored in said second means for storing.

16. The computer interface defined in Claim 12, further comprising counting means for identifying received information associated with overlapping areas of sensitivity.

17. A computer interface comprising:

an input for receiving information from a first computer environment;

a memory for storing said received information;

a plurality of predetermined rules, said rules for logically translating the stored information to information for use in a second computer environment;

a loop routine for selectively testing said plurality of rules to determine if any of said rules should be implemented; and

an output for communicating a signal representative of the translated information.

18. The computer interface defined in Claim 17, wherein the loop routine further comprises means for retrieving from said memory information associated with a first rule if said first rule is determined to be implemented.

19. The computer interface defined in Claim 17, wherein each rule comprises a subroutine of operations necessary to translate said stored information.

20. The computer interface defined in Claim 17, wherein said memory stores said received information in a form representative of the manner said received information appears on a display screen.

21. The computer interface defined in Claim 17, wherein said signal representative of the translated information is communicated to said second computer environment.

22. The computer interface defined in Claim 17, further comprising means for identifying a subset of rules from the plurality of rules to be implemented in response to said received information, wherein said means for identifying tests predetermined areas of sensitivity associated with said subset of rules.

23. The computer interface defined in Claim 22, wherein said means for identifying a subset of rules further comprises means for flagging rules determined to be implemented from the plurality of rules.

24. The computer interface defined in Claim 23, further comprising means for storing a representation of the areas of sensitivity of said flagged rules.

25. The computer interface defined in Claim 24, wherein a signal indicative of received information associated with a first area of sensitivity is stored in said means for storing.

26. The computer interface defined in Claim 25, further comprising counting means for identifying received information associated with overlapping areas of sensitivity.

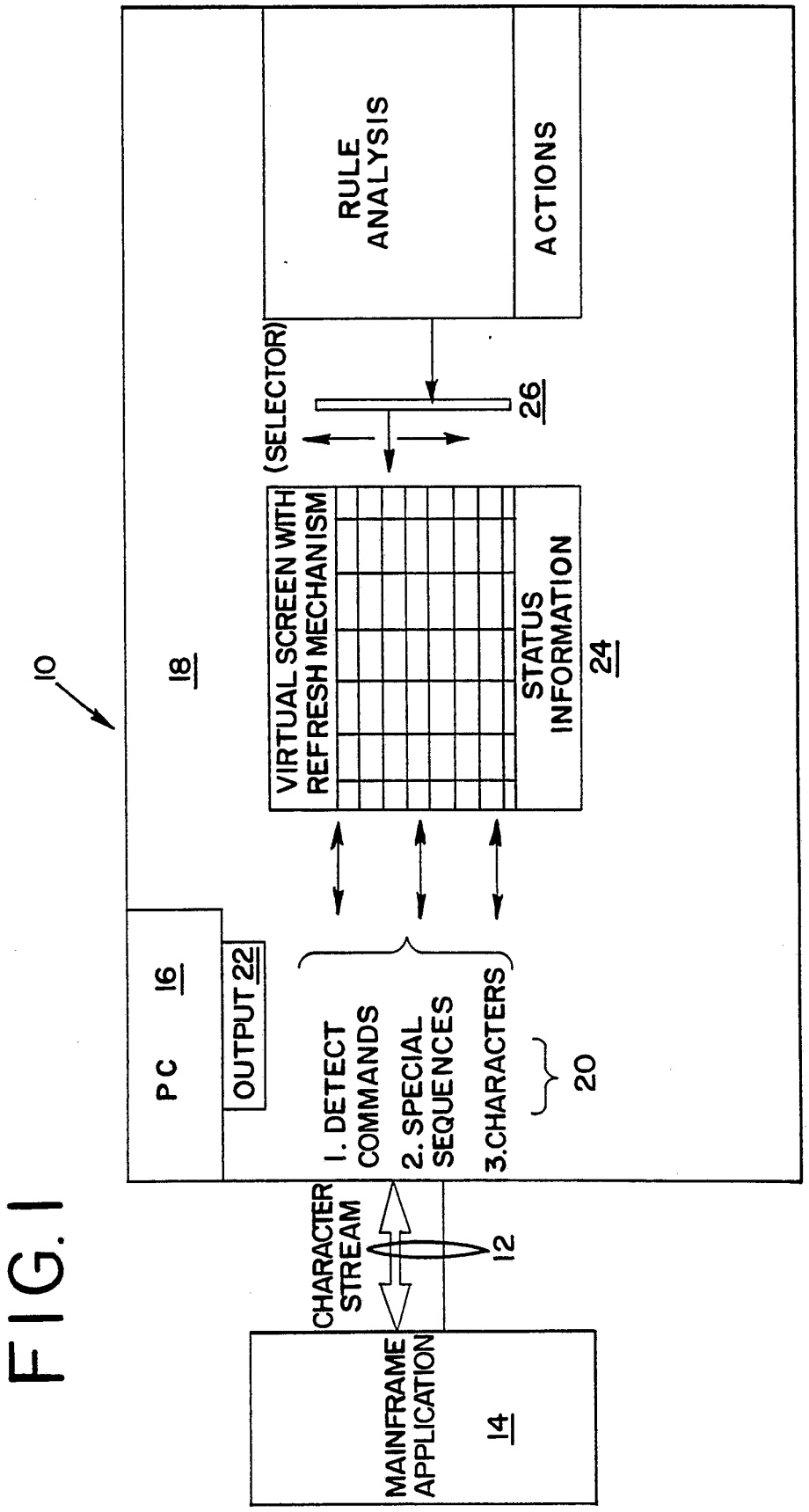

10
18
RULE ANALYSIS
ACTIONS
(SELECTOR)
26
VIRTUAL SCREEN WITH REFRESH MECHANISM
STATUS INFORMATION
24
PC 16
OUTPUT 22
1. DETECT COMMANDS
2. SPECIAL SEQUENCES
3. CHARACTERS
20
CHARACTER STREAM
12
MAINFRAME APPLICATION
14

FIG. 1

FIG. 2

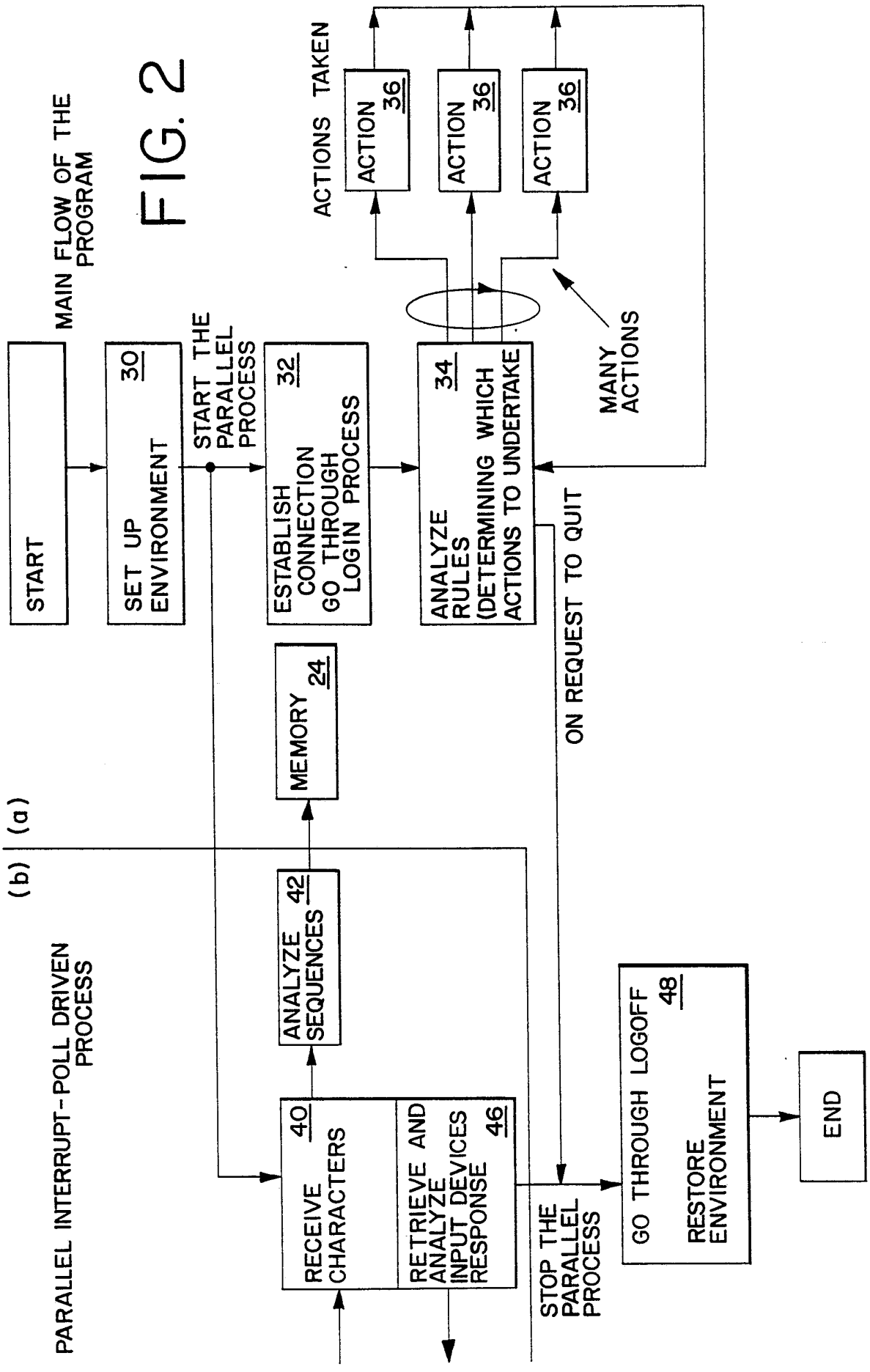

MAIN FLOW OF THE PROGRAM
START
SET UP ENVIRONMENT 30
START THE PARALLEL PROCESS
ESTABLISH CONNECTION GO THROUGH LOGIN PROCESS 32
ANALYZE RULES (DETERMINING WHICH ACTIONS TO UNDERTAKE 34
ACTIONS TAKEN
ACTION 36
ACTION 36
ACTION 36
MANY ACTIONS
ON REQUEST TO QUIT
MEMORY 24
(a)
(b)
PARALLEL INTERRUPT-POLL DRIVEN PROCESS
ANALYZE SEQUENCES 42
RECEIVE CHARACTERS 40
RETRIEVE AND ANALYZE INPUT DEVICES RESPONSE 46
STOP THE PARALLEL PROCESS
GO THROUGH LOGOFF 48
RESTORE ENVIRONMENT
END

FIG.3

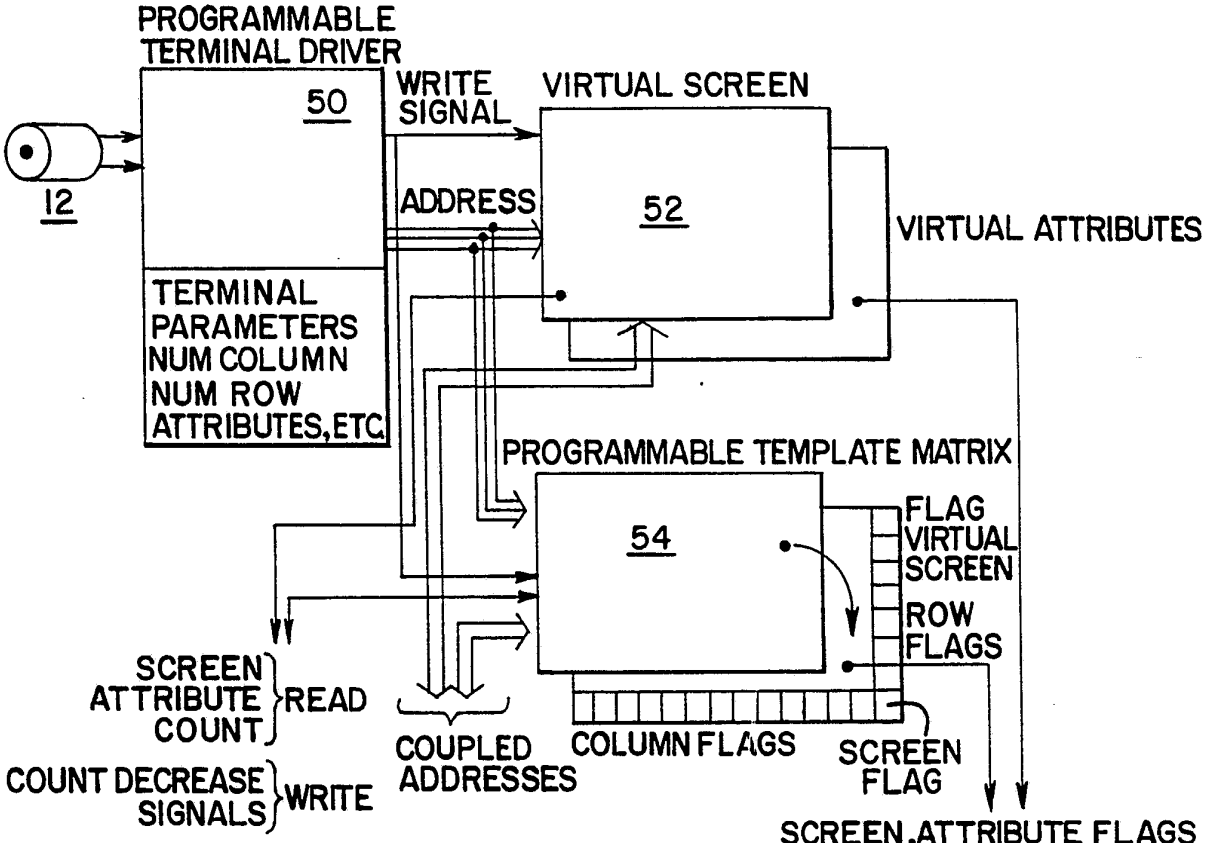


FIG.4A

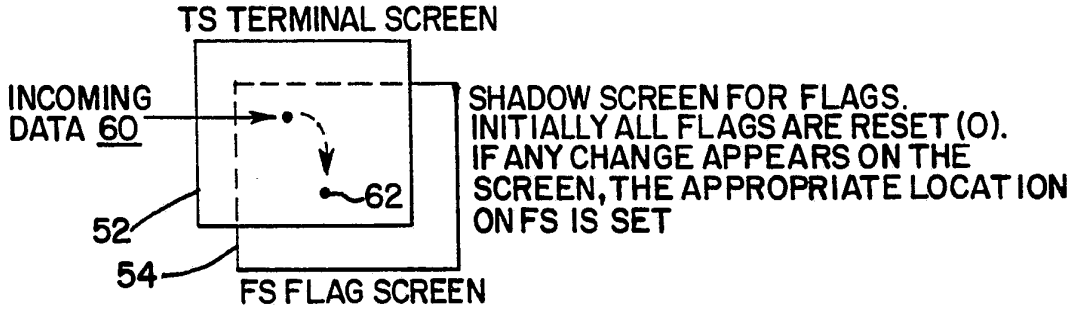


FIG.4B

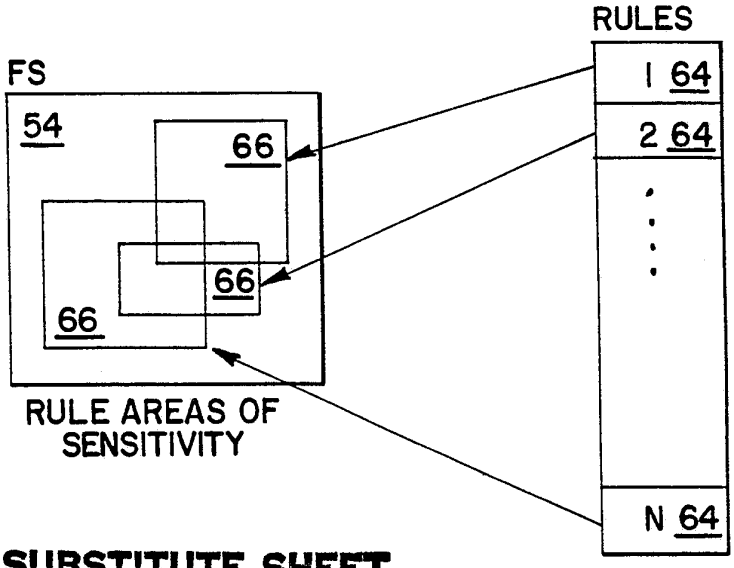


FIG.5

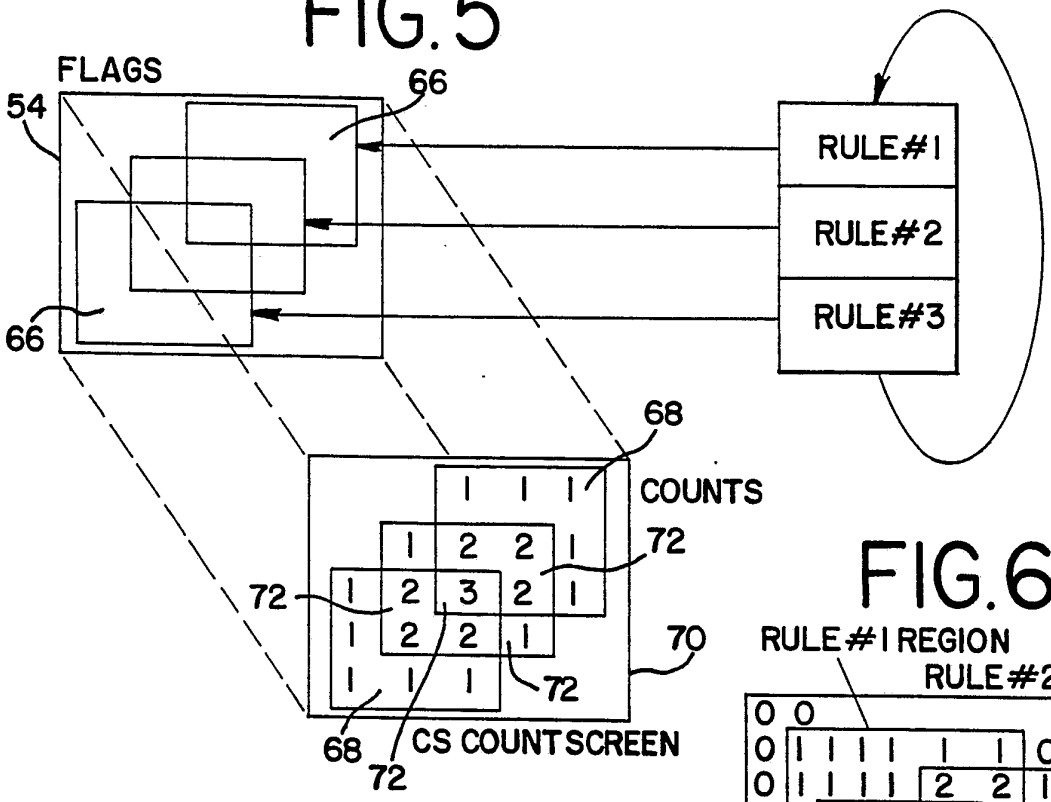


FIG.6

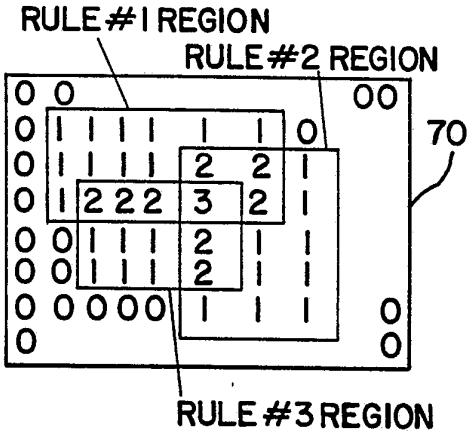
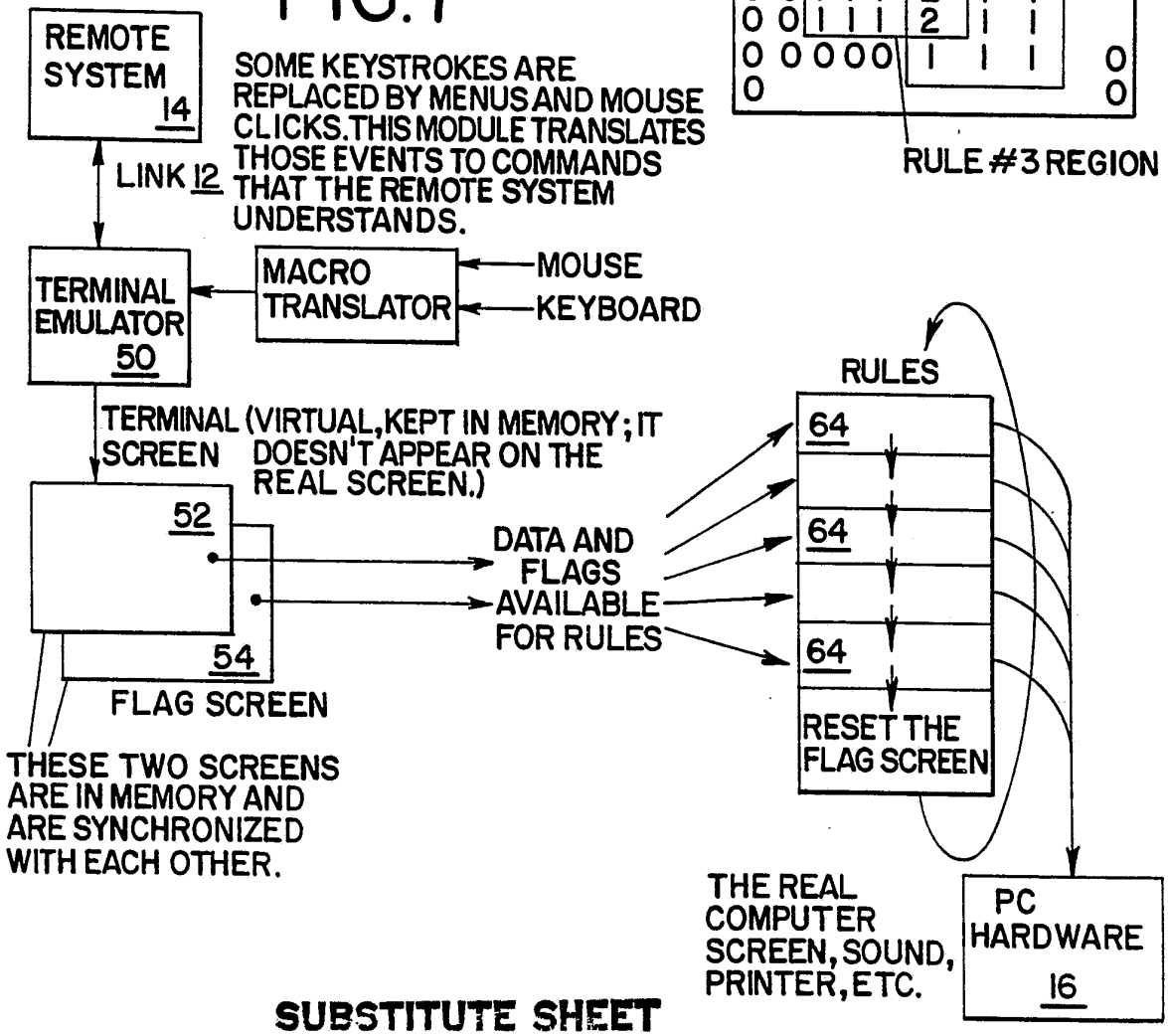


FIG.7



INTERNATIONAL SEARCH REPORT

PCT/US93/03258

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06F 13/00, G06F 15/16
US CL :395/500
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/500 395/ALL

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS US Pat

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A, 4,628,446 (Hoffner) 09 December 1986 Figs. 1,2, Abstract, Col. 2 and 3.	1-26
X	US,A, 4,281,315 (Bauer) 28 July 1981 Figs. 1,2 Abstract Col. 3-12.	1-26

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 25 MARCH 1993	Date of mailing of the international search report 07 JUN 1993
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE	Authorized officer DAVID Y. ENG Telephone No. (703) 305-9691

Form PCT/ISA/210 (second sheet)(July 1992)*