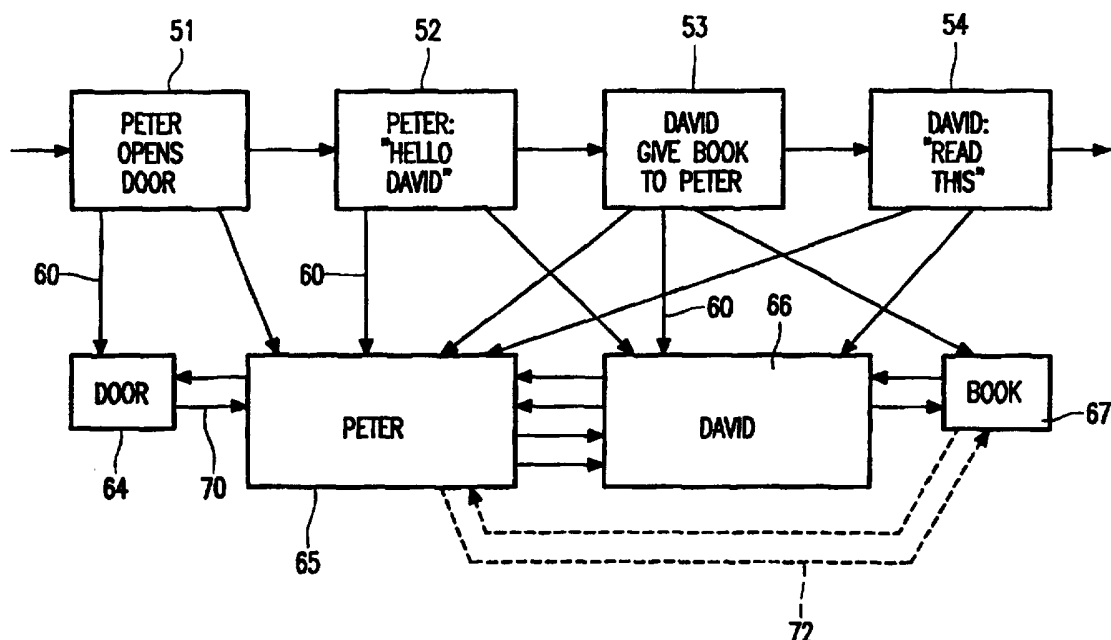




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>6</sup> :</b> <b>G06F 9/44</b>	<b>A2</b>	<b>(11) International Publication Number:</b> <b>WO 97/42571</b> <b>(43) International Publication Date:</b> 13 November 1997 (13.11.97)
<b>(21) International Application Number:</b> PCT/IB97/00300 <b>(22) International Filing Date:</b> 25 March 1997 (25.03.97)  <b>(30) Priority Data:</b> 9609189.7 2 May 1996 (02.05.96) GB  <b>(71) Applicant:</b> PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL). <b>(71) Applicant (for SE only):</b> PHILIPS NORDEN AB [SE/SE]; Kottbygatan 7, Kista, S-164 85 Stockholm (SE). <b>(72) Inventors:</b> WAVISH, Peter, Roy; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). CONNAH, David, Malcolm; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). <b>(74) Agent:</b> WHITE, Andrew, G.; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).		<b>(81) Designated States:</b> JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).  <b>Published</b> <i>Without international search report and to be republished upon receipt of that report.</i>

(54) Title: ADAPTIVE PROCESS MODELLING AND CONTROL



## (57) Abstract

A rule-based data processing system includes a memory having stored therein a number of groups of agents, each group being defined by respective sets of rules, with the system running the respective sets of rules in parallel. The agents are in two clusters, the first being "script agents" (51-54) which are arranged as a sequentially enabled chain, and "operator agents" (64-67) arranged to carry out selected procedural steps. The selection of the step is made within each operator agent (65) on the basis of one or more external factors, and is triggered by the enablement of a script agent (51) of the sequence prior to that script agent (52) carrying a marker (60) identifying the operator agent.

**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.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon	KR	Republic of Korea	PL	Poland		
CN	China	KZ	Kazakhstan	PT	Portugal		
CU	Cuba	LC	Saint Lucia	RO	Romania		
CZ	Czech Republic	LI	Liechtenstein	RU	Russian Federation		
DE	Germany	LK	Sri Lanka	SD	Sudan		
DK	Denmark	LR	Liberia	SE	Sweden		
EE	Estonia			SG	Singapore		

## DESCRIPTION

**ADAPTIVE PROCESS MODELLING AND CONTROL**

5           The present invention relates to rule-based data processing apparatuses and in particular to those having stored therein one or more agents, the or each agent being defined by a set of rules, and each rule being in the form of a condition portion and an associated response portion, with the condition portion of each rule requiring the presence of one or more specified behaviour states  
10       or external inputs and the apparatus comprising means for responding to any said condition being satisfied by generating the associated response. The invention additionally, but not exclusively, relates to process control and interactive entertainment apparatuses based around such a rule-based data processing apparatus.

15           Rule-based processor systems may support behaviour languages in which behaviours are either simple operations performed on a value, or rules, where a rule defines logical and temporal relations between events (behaviour state changes). Events associated with the condition portion of a rule - also  
20       known as the left hand side behaviours of a rule - cause subsequent events associated with the response portion (right hand side behaviour) of the rule. An example of such a rule-based real time language is Real Time ABLE (RTA), described in "Simulating and Implementing Agents and Multiple Agent Systems" Proceedings of the European Simulation Multiconference 1991 at pages 226-  
25       231. ABLE stands for Agent Behaviour Language and is a highly concurrent production rule language for simulating agents and multiple agent systems, and provides for tighter integration of time with the production rule system. A further example of a rule-based language is given in "The Behaviour Language; User's Guide" by R A Brooks, AI Memo 1227, Massachusetts Institute of  
30       Technology Artificial Intelligence Laboratory, April 1990.

          Whilst rule-based languages provide greater flexibility than more

traditional programming languages, their propagating nature can still lead to rigidity in, for example, process control where activation of process stages is strictly controlled in terms of starting, intermediate and finishing rule conditions and responses with little flexibility for unprogrammed events. Modification of  
5 the process to accommodate even minor procedural changes is also not a simple matter, sometimes requiring recompilation of the whole program to accommodate a small change in a programmed delay or procedure start time.

Rule-based systems have been suggested as particularly suited for artificial intelligence applications where more complex behavioural simulations  
10 may be implemented in interconnected networks or modules of agents. The above-mentioned problems recur however when modelling interaction scenarios between virtual "actors", each in the form of a compound group of interconnected agents, with different agents governing the virtual actors response to environmental factors and the actions of other virtual actors. If a  
15 "script" is to be followed, all of the virtual actors required conditions and potential responses must be programmed into the script such that the interaction becomes a rigidly controlled process with little or no variation in successive runnings. From an observers point of view, the 'interaction' between the virtual actors is predictable and unrealistic.

20 It is therefore an object of the present invention to provide a configuration of data processing apparatus which may provide the basis for more flexible control applications.

It is a further object of the invention to provide a means for process  
25 control, based on such a data processing apparatus, which is readily adaptable to changes in procedure.

It is yet a further object of the present invention to provide an interactive entertainment apparatus where characters modelled as compound agents may interact with each other whilst following a script in a relatively flexible and  
30 realistic manner.

In accordance with the present invention there is provided a rule-based data processing apparatus comprising a processor coupled with memory means having stored therein a first and at least one further group of agents, each group being defined by respective sets of rules, and the processor being  
5 arranged to run said respective sets of rules in parallel, characterised in that the agents of the first group (hereinafter "script agents") are arranged as a chain being enabled in sequence, said at least one further group of agents (hereinafter an "operator agent") is arranged to carry out a selected one of two or more procedural steps, with the selection being made within the operator  
10 agent on the basis of one or more external factors, and the selection operation within the operator agent is triggered prior to the enablement of a script agent carrying a marker identifying the operator agent, and subsequent to the enablement of a preceding script agent of the sequence.

By providing the different classes of agent, the script or procedure  
15 defined by the string of script agents may be modified without affecting the operator agents. Similarly, the operator agents may be modified in terms of the skill or process they perform without amendment of the script which simply calls for the skill at particular times. Quick substitution of operator agents with different skills or skill levels provides the system with a high degree of flexibility.  
20 The apparatus may comprise two or more operator agents with at least one of the said external factors (on the basis of which one of the operator agents makes the said selection), being the current setting of a controllably variable value set by another of the operator agents. Thus, not only are the operator agents arranged to interact with the progressive string of script agents, they are  
25 also configured to interact with each other.

The or each of the script agents may be configured to identify when the or each operator agent for which it carries a marker has made the said selection following triggering by the enablement of a preceding script agent, and may be configured to modify an internal behaviour value on making said  
30 identification. By setting of this internal behaviour value, which may be a simple flag, it becomes a relatively simple matter to determine the progress of

the process or script. Additionally, based on this internal monitoring, each of the script agents may suitably be configured to unload itself from the memory means following its enablement and the said modification of internal behaviour value, in other words determining when the process stage it is directing has been carried out and then cancelling itself as redundant.

One or more of the script agents may include an internal mechanism operable to generate a first representation of factors external to the script agent and referred to by functions supported by that script agent, and to generate a second representation of said functions supported by the script agent, to compare the said first and second representations, and to generate and output an indication of any inconsistency between said first and second representations. This local diagnostic feature would enable a script agent to compare its own input/output conditions with output/input conditions specified for neighbouring script agents to check for mismatches. Such a facility would be of particular use where a sequence of script agents was to be edited: the facility would reduce the necessity for recompilation of the script agents as a whole.

According to one embodiment of the invention, the above-described rule-based data processing apparatus is provided with an external interface means coupled with the data processor thereof to provide a process control apparatus, with said external interface means being configured to control a plurality of external processes linked thereto on triggering of respective operator agents.

According to another embodiment of the present invention, the above-described rule-based data processing apparatus is configured as an interactive entertainment apparatus, having output means coupled with the data processor and arranged to generate an indication of the operation of the or each of said operator agents. In such an embodiment, the output means may suitably be means operable to generate an image of a virtual environment and to animate one or more modelled features in said environment, with the animation of the or each of said one or more modelled features being controlled by a respective operator agent. At least one further agent or group of agents may be provided

within the virtual environment (for example controlling environmental artifacts and/or unscripted characters), with the or each said further agent being triggered by actions initiated by one or any of said operator agents.

Further features and advantages of the present invention are recited  
5 hereinafter and in the attached claims, to which reference should now be made.

Preferred embodiments of the present invention will now be described, with particular reference to RTA agents by way of example only, and with reference to the accompanying drawings, in which:

10 Figure 1 is a block schematic diagram of a rule-based data processing apparatus according to an embodiment of the present invention;

Figure 2 is a schematic diagram illustrating a sequence of script agents linked to associated operator agents; and

Figure 3 is a timing diagram illustrating the operational inter-relationship  
15 of a number of script and operator agents.

Figure 1 shows apparatus embodying the present invention, comprising a processor 10 operating on a compiled asynchronous logic circuit representation defined by a population of RTA rules which, through an  
20 interfacing link 8, is enabled to control an application such as an industrial process or multi-character interactive animation. The specific features of the application controlled via the interfacing link are not required for an understanding of the operation of the present invention and, except where such an application serves to illustrate the operation, they will not be described  
25 further.

The processor 10 has associated with it a random access memory (RAM) 12 which provides a number of discrete storage areas, a first of which 14 holds a table containing details of the various elements of the asynchronous logic circuit representation of the compiled population of RTA rules, with  
30 discrete areas of the table holding the details for script agents 14A and for each of the operator agent groups (in this example three) 14B, 14C, 14D, as

will be described. The compilation of RTA rules to an asynchronous logic circuit representation is described in greater detail in our European patent application EP-A-0704077 (PHB 33903) the disclosure of which is incorporated herein by reference.

5 Each of the behaviours (register elements), time annotations (delay elements) logic function elements and rules forming the asynchronous logic circuit representation has a row in the table 14. Each row in the table identifies the element number, a state flag of the device, a queued status flag of the device, the internal status, and a propagate function start address. When the  
10 state of the element to which the row of the table relates changes (referred to herein as an event), a propagate function is executed to effect any appropriate changes to all of the elements which may be affected by such a change. These propagate functions are conveniently arranged at certain memory locations or start addresses. Changing the state of elements by propagation  
15 in such a manner rather than by searching for all of the elements affected by an event allows the apparatus to operate efficiently.

The propagate functions starting at the start addresses are stored in another area 16 of the RAM 12; the RAM also provides further storage areas for future element state changes 18 and an event stack 20. As with the table  
20 14, each of the memory areas 16, 18, 20 is divided into discrete areas for the script agents (suffix A) and for the different operator agents (suffix B, C, D). A program store 22 holds the command structure governing the compilation of the asynchronous logic circuit for each of the agents, script or operator. The program store 22 and the propagate functions 16 could, if desired, be stored  
25 in another memory device, for example a read-only memory.

The RAM 12 is connected to the processor 10 by a data bus 24 and an address bus 26 in known manner. Also in known manner, a clock (CLK) 28 is connected to the processor. The processor 10 operates on the compiled representation according to the command structure from program store 22 with  
30 propagation of state changes through the representation equating to responding to the satisfaction of condition behaviours by generating one or more further



behaviours defining the associated response as specified by the rule populations which define the compiled circuit representation. Optionally, a user interface device (UID) 30 may be provided to enable input of control data and commands directly to the processor 10, if required by the application.

5           The storage area 18 for future agent element state changes is arranged, for the script agents and for each of the operator agent groups, as a two row table containing a plurality of time slots and element numbers. Any number of elements numbers can be stored to correspond to a particular time slot and the states of these elements will all be changed during that time slot. The table  
10       may be arranged for a consecutive sequence of time slots, but it is preferred to use a stack containing only those time slots for which element numbers are stored, with the facility to insert element numbers and time slots as required due to operation of the program.

          Generally, all of an agent's element state changes except those  
15       corresponding to delay elements and behaviour state changes will be executed in the current time slot although other element state changes can be also delayed if desired. Those element state changes which are to occur without a time delay may be placed in the event stack 20 for execution in the near future - i.e. later in the current time slot, but before moving to the next time slot  
20       for which an event is specified.

          A particular use for the process control provided by the present invention is in interactive entertainment or artificial intelligence applications. This addresses the problem of how to create a performance of a scripted scenario using a set of operator agents as virtual actors playing, for example, characters  
25       in a computer game. The virtual actors are required to perform a script autonomously, as real actors do, rather than being directly controlled by it. In accordance with an aspect of the present invention, the script is represented at run-time as a set of autonomous agents and so plays an active part in the operation of the system, with the virtual actors being distinct groupings of  
30       agents whose activity is partly determined by the situation in which they are embedded (and which their activity is helping to construct) and partly by the

script which they enact.

As with theatrical rehearsals or read-throughs, the control of the performance of the script is distributed amongst the virtual actors, each of whom is responsible for identifying the sentence in the script to be performed and for identifying the cue for performing it - typically when the previous sentence has been performed, where a sentence may correspond to a speech by an actor, or a stage direction.

The individual or grouped operator agents comprising the virtual actors, being controlled by simple situated agents, require the script to be translated into a form that is "intelligible to" (capable of implementation by) them. To achieve this, the written script is translated into a linked chain of script agents, each one of which is concerned with, and represents, a particular sentence in the script. Referring back to Figure 1, the data for these script agents is held in the area 14A of the table 14, with the further discrete areas of the table 14B, 14C, 14D holding the details for the respective virtual actors. As will become clear, each virtual actor may comprise a highly complex arrangement of interlinked agents governing not only how the virtual actor is to physically handle a task, but also behavioural aspects which may in turn affect the physical operation. Other virtual actors however may be little more than theatrical "props" such as a door which may be open or closed or a book which may be passed from one character to another.

As schematically illustrated in Figure 2, each script agent 51-54 contains within it markers 60 on the actor performing it, and on the other actors and agents that are involved in its performance. The Figure represents a brief extract from a scripted exchange between two virtual actors (Peter and David) during the course of which Peter opens a door (to a room where David is) and greets him: David responds by handing a book to Peter and telling him to read it. This is broken down into four script agents 51-54 with respective markers to the virtual actor/operator agents 64-67 representing respectively a door 64 (to the room where David is), Peter 65, David 66 and the book 67. The first script agent 51 ("Peter opens door") has markers 60 on both Peter 65 and the

door 64, which Peter may be required to open using internal skills and in a "style" determined by others of his internal characteristics. The exact form of interaction between the 'Peter' and 'door' virtual actors (represented by links 70) is not explicitly scripted but instead merely initiated by the script.

5           The second script agent 52 (Peter says "Hello David") has markers to both the 'Peter' agent 65 and the 'David' agent 66. Whilst David does not have a direct role from script agent 52, his presence within the room is required if Peters utterance is to have any meaning; the completion of actions associated with the script agent 52 is also the cue for the actions of David as scripted in  
10       the following script agent 53 ("David gives book to Peter"), which script agent has markers 60 on the agents for David 66, Peter 65 and the book 67. In terms of the interaction between the virtual actors, that between Peter and the door (represented by links 70) ceases when the storyline moves to script agent 52 which does not have a marker on the door 64, whilst the interaction between  
15       Peter and the book (represented by dashed links 72) is not established until script agent 53 is reached which has markers on both Peter and the book. Script agents also have additional internal state and behaviour, described below, which may for instance register whether the action that they represent has been performed.

20           In operation, each of the virtual actors performs the script by searching forwards through the chain of script agents for the next action it is to perform, and searching backwards from that point to find the script agent corresponding to the cue for its next action, generally (but not necessarily) the preceding script agent. From the markers held by the two script agents it is a relatively simple  
25       matter for the virtual actor agents to determine the virtual actor from whom it takes its cue, and the virtual actors and objects with respect to which its action needs to be performed.

          The particular advantage of implementing the script agents as autonomous agents is that a number of extra system features become easy to  
30       implement by adding extra behaviour to the script agents. For example, the script agents can monitor their own performance, either by a behaviour

mechanism directly noting when the script segment has been performed, or by monitoring the virtual environment for particular conditions or conjunctions of conditions satisfying the starting conditions for the next script agent.

Script agents can prompt or direct the actor performing them by directly  
5 asserting a specific condition or conditions, and similarly can cue the next actor agent. By use of an established RTA mechanism, the script agents can unload themselves after they have been performed in order to save memory space.

To enhance the system functionality, the script and actor agents may be arranged to collectively display a time map of the performance by individually  
10 drawing time-bars corresponding to their own individual performance as shown in Figure 3, where traces A to E represent script agents and traces F to H represent virtual actors. The display device (not shown) may be a part of the UID 30 (Fig.1) or more suitably may be driven via the interfacing link 8. This graphical user interface provides a way of communicating with the director or  
15 designer concerning the agents content or performance, and can provide a means of previewing or checking the script by propagating a "present moment" signal down the chain of fragments (along the t-axis) delayed in time by each agent to correspond with the expected duration of the agent's performance.

The individual script agents are suitably configured to check for  
20 consistency between themselves and their neighbours, for example comparing cues against markers held by the next script agent, such as to provide an internal diagnostic function for modified or inserted sections of the script. If suitably configured, the script agents may be arranged to generate reports on the sentences in the written script they carry, with annotations representing  
25 desired modifications to their performance. Such annotations might, for example, indicate that it was not the only sentence being spoken at the time, thereby indicating that more time should be allowed for the speech from the preceding script agent. In a further development, the script agents may be operable to develop additional script agents to fill gaps in the script: for  
30 example, a consistency check between a sequential pair of script agents shows a character to be in two different locations - this may lead to the generation of

an additional script agent between the existing pair requiring the character to travel from the first to the second location.

By enabling the script at run-time as a set of autonomous agents, rather than as static objects, the additional functionality described above may be simply implemented. In order to further enhance the functionality, steps may be taken to deal with referential expressions in the script such as "the red book on the table" which at the time the script object is generated may have no references. A proposed solution is to have another set of autonomous agents which stand in for the objects referred to, and so can be pointed to by markers in the script agents, but can also autonomously connect themselves to the objects and places that they represent in the simulated 3-D world in which the script is performed. This technique is covered in greater detail in our co-pending United Kingdom Patent Application No 9609197.0.

From reading the present disclosure, other modifications will be apparent to persons skilled in the art. Such modifications may involve other features which already known in the field of data processing apparatuses and component parts thereof and which may be used instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure of the present application also includes any novel feature or any novel combination of features disclosed herein either explicitly or implicitly, whether or not it relates to the same invention as presently claimed in any claim and whether or not it mitigates any or all of the same technical problems as does the present invention. The applicants hereby give notice that new claims may be formulated to such features and/or combinations of such features during the prosecution of the present application or of any further application derived therefrom.

## CLAIMS

1. A rule-based data processing apparatus comprising a processor coupled with memory means having stored therein a first and at least one further group of agents, each group being defined by respective sets of rules, and the processor being arranged to run said respective sets of rules in parallel, characterised in that the agents of the first group (hereinafter "script agents") are arranged as a chain being enabled in sequence, said at least one further group of agents (hereinafter an "operator agent") is arranged to carry out a selected one of two or more procedural steps, with the selection being made within the operator agent on the basis of one or more external factors, and the selection operation within the operator agent is triggered prior to the enablement of a script agent carrying a marker identifying the operator agent, and subsequent to the enablement of a preceding script agent of the sequence.

2. Apparatus as claimed in Claim 1, comprising two or more operator agents wherein at least one of said external factors, on the basis of which one of the operator agents makes said selection, is the current setting of a controllably variable value set by another of the operator agents.

3. Apparatus as claimed in Claim 1 or Claim 2, wherein each of said script agents is configured to identify when the or each operator agent for which it carries a marker has made said selection following triggering by the enablement of a preceding script agent, and to modify an internal behaviour value on making said identification.

4. Apparatus as claimed in Claim 3, wherein each of said script agents is configured to unload itself from said memory means following enablement and said modification of said internal behaviour value.

5. Apparatus as claimed in any of Claims 1 to 4, wherein one or

more of said script agents includes an internal mechanism operable to generate a first representation of factors external to said script agent and referred to by functions supported by said script agent, to generate a second representation of said functions supported by said script agent, to compare said first and second representations, and to generate and output an indication of any inconsistency between said first and second representations.

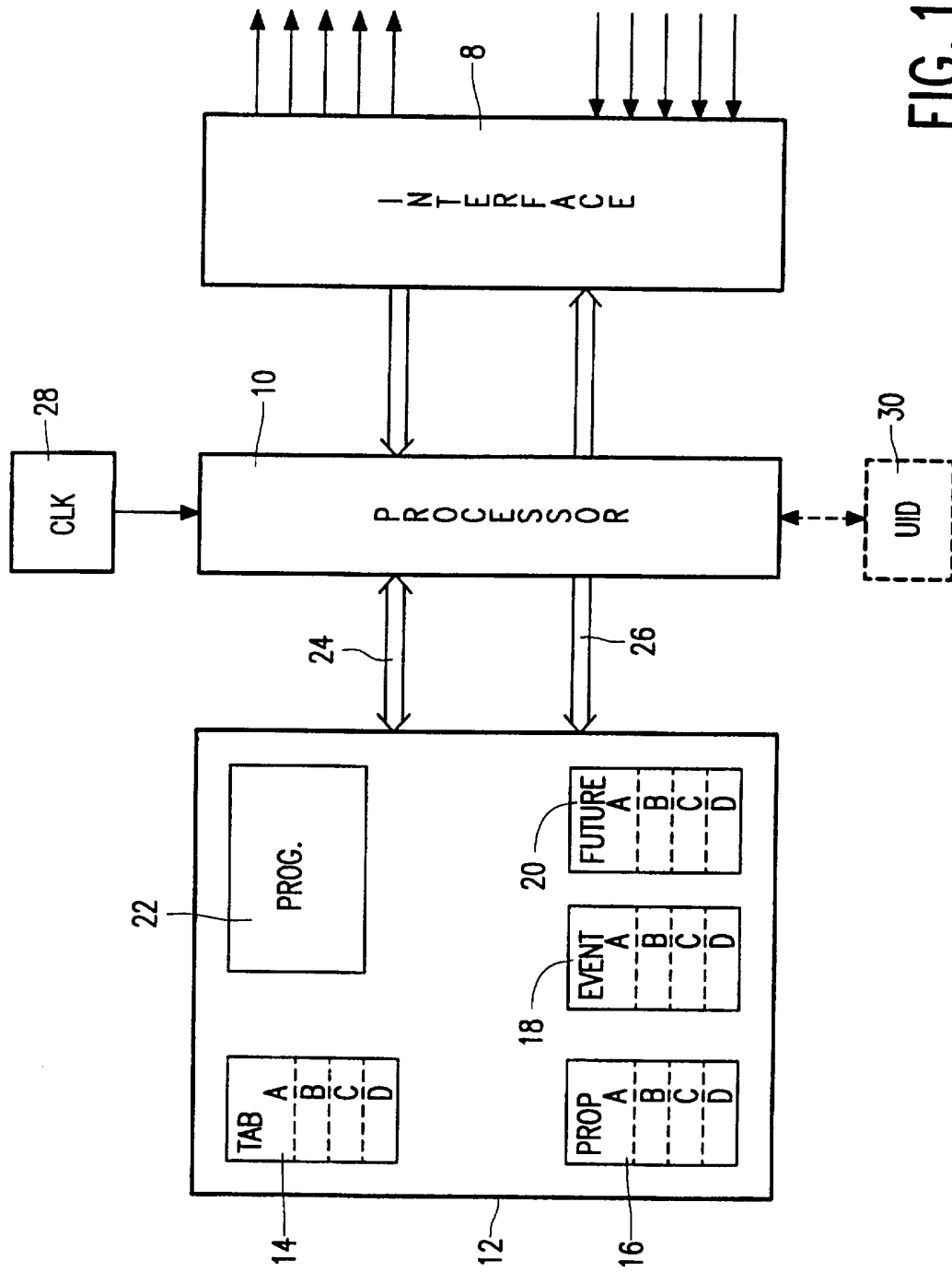
6. A process control apparatus comprising a rule-based data processing apparatus as claimed in any of Claims 1 to 5, and external interface means coupled with the processor thereof, said external interface means being configured to control a plurality of external processes linked thereto on triggering of respective operator agents.

7. Interactive entertainment apparatus comprising a rule-based data processing apparatus as claimed in any of Claims 1 to 5, and output means coupled with the processor thereof and arranged to generate an indication of the operation of the or each of said operator agents.

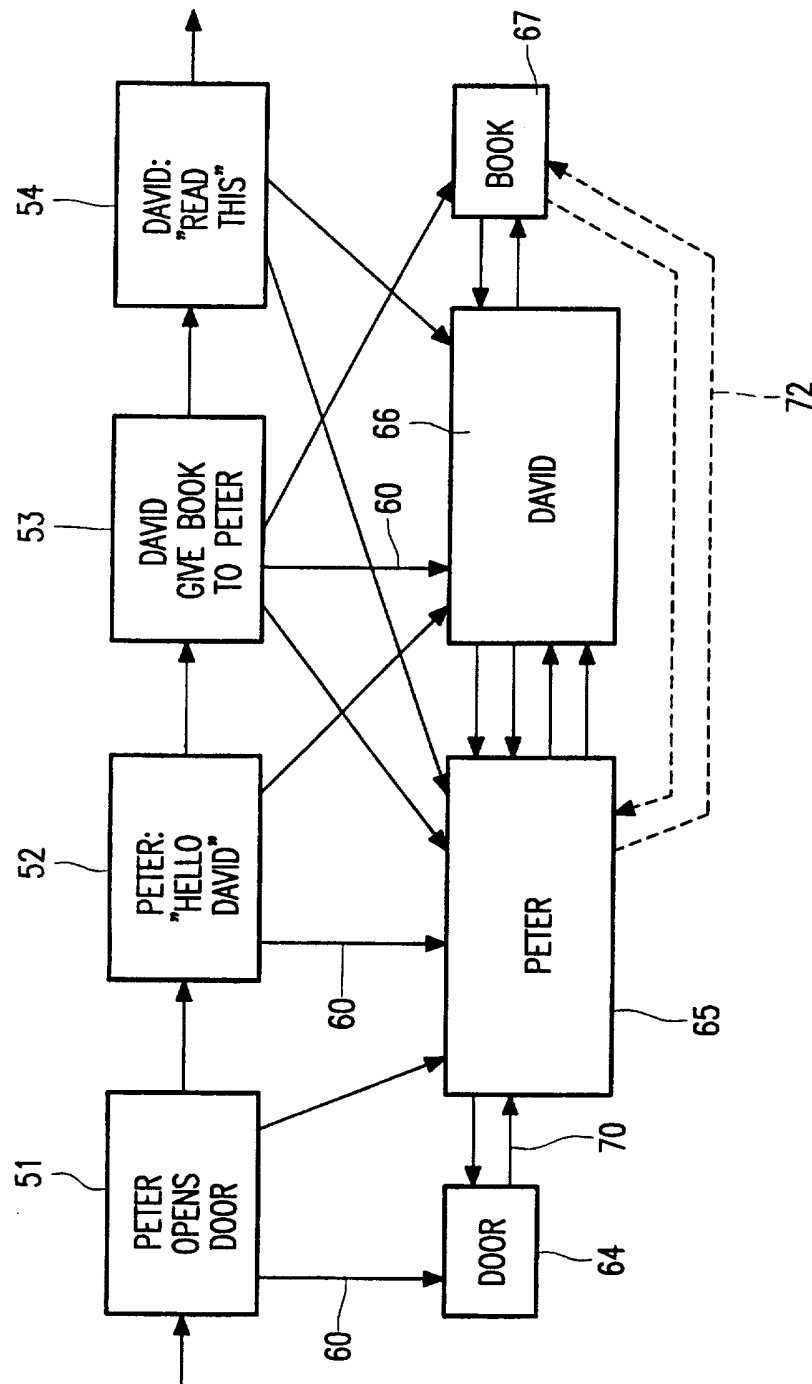
8. Apparatus as claimed in Claim 7, wherein said output means comprises means operable to generate an image of a virtual environment and to animate one or more modelled features in said environment, the animation of the or each of said one or more modelled features being controlled by a respective operator agent.

9. Apparatus as claimed in Claim 8, further comprising at least one further agent or group of agents, the or each said further agent being triggered by actions initiated by one or any of said operator agents.

1/3







**FIG. 2**

3/3

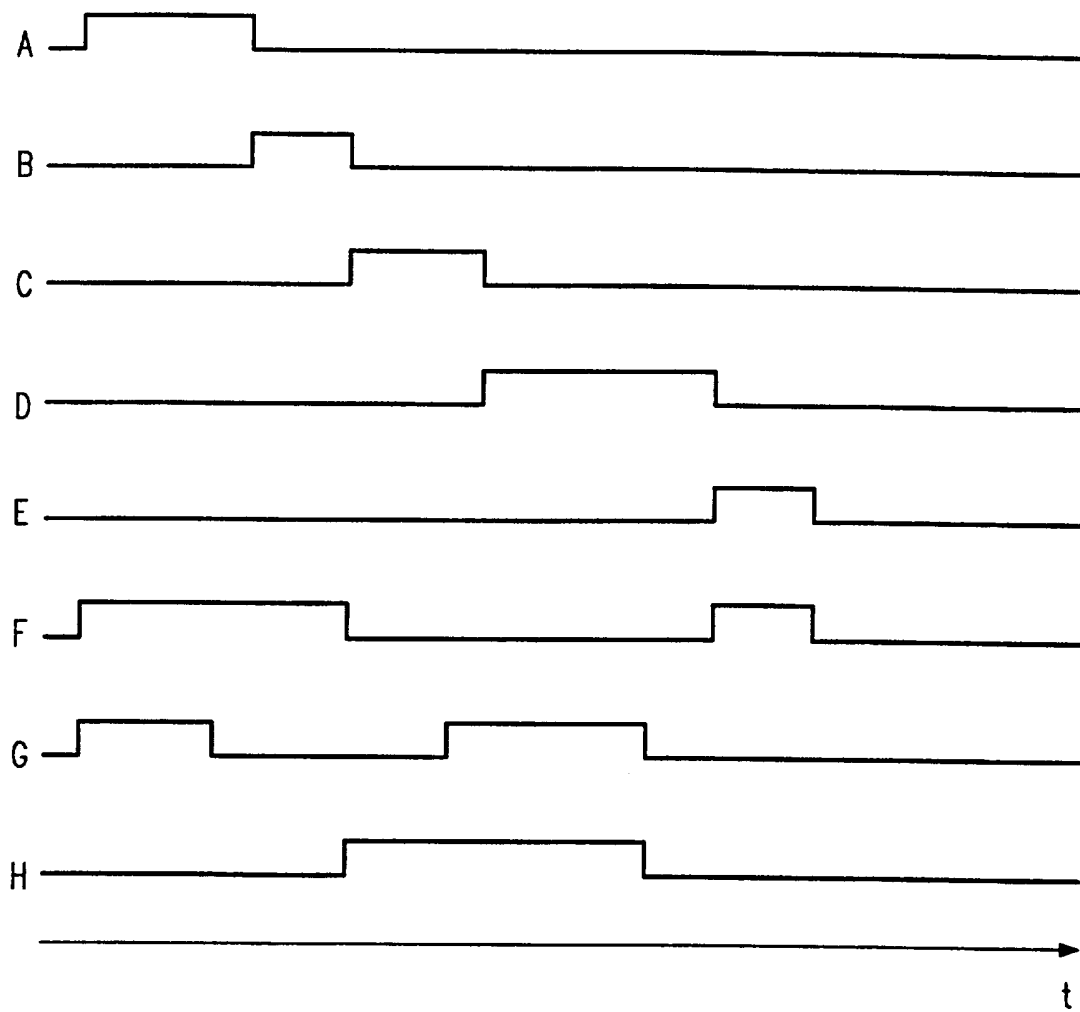


FIG. 3