

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
29 April 2010 (29.04.2010)

PCT

(10) International Publication Number  
**WO 2010/045733 A1**

(51) International Patent Classification:

G06F 17/20 (2006.01) G06T 13/00 (2006.01)  
G06F 17/27 (2006.01) G06T 15/70 (2006.01)

(21) International Application Number:

PCT/CA2009/001518

(22) International Filing Date:

22 October 2009 (22.10.2009)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/107,541 22 October 2008 (22.10.2008) US

(71) Applicant (for all designated States except US): **XTRA-NORMAL TECHNOLOGY INC.** [CA/CA]; 5555 Casgrain avenue, 2nd Floor, Montreal, Québec H2T 1Y1 (CA).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BHERER, Hans** [CA/CA]; Montreal, Québec (CA). **LANGE, Herve** [CA/CA]; Montreal, Québec (CA).

(74) Agent: **BENOIT & COTE S . E . N . C .** ; 1001 de Maisonneuve Blvd. West, Suite 210, Montreal, Québec H3A 3C8 (CA).

(81) Designated States (unless otherwise indicated, for every kind of national protection available):

AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available):

ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: IMPLICIT ACTIONS GENERATOR IN TEXT-TO-ANIMATION SYSTEMS

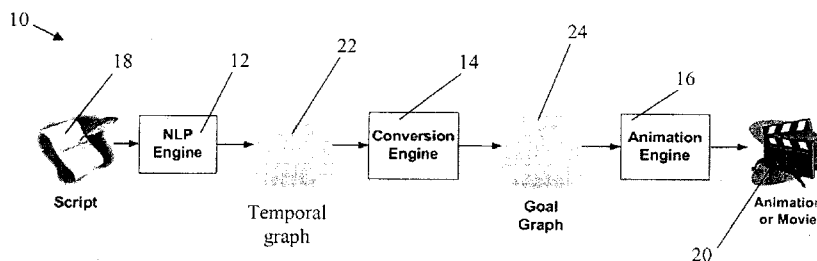


Fig. 1

(57) Abstract: There is described a method and system for converting text for input into an animation generator. The method comprises associating an explicit goal for each explicit action in a text, determining an un-achievability of the explicit goal based on an initial state, upon determining the un-achievability, generating at least one implicit goal associated with an implicit action, the implicit action changing the initial state -when executed to render the explicit goal achievable, temporally organizing the explicit goal and the at least one implicit goal together in a goal temporal organization and generating the animation based on the goal temporal organization, the animation representing the at least one explicit action associated -with the explicit goal and the implicit action associated with the at least one implicit goal.



WO 2010/045733 A1

## **IMPLICIT ACTIONS GENERATOR IN TEXT-TO-ANIMATION SYSTEMS**

### **CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] This application claims priority from US provisional patent application 61/107,541 filed October 22, 2008 and entitled "IMPLICIT ACTIONS GENERATOR IN TEXT-TO-ANIMATION SYSTEMS".

### **TECHNICAL FIELD**

[0002] The present disclosure relates to Text-to-movie (TTM) and text-to-animation (TTA), and more particularly to the natural language processing and implicit actions in a text.

### **BACKGROUND**

[0003] Text-to-movie (TTM) and text-to-animation (TTA) systems convert a text written in a natural language into a movie or an animation, respectively. The movie or the animation reflects what is being described in the text. However, a description written by a user and inputted in a TTM or TTA system can be incomplete, which means that there are missing actions in the input text in order to properly render the intent of the user. A user often omits some actions because he considers them as being obvious or implicit. For example, a user inputs the following text: "Paul sits on a chair. Paul walks to the door". An animation engine cannot properly render an animation corresponding to this description since there is a missing action to link the two sentences of the input text. Indeed, Paul cannot walk from a sitting position. Implicitly the user knows that Paul first stands up before walking. However, an animation engine does not have the common sense knowledge of human beings, and therefore, it cannot automatically insert the proper intermediate actions.

[0004] Inserting proper actions to correct the problem is not trivial: software that attempts an automatic "fix" is generally based on a planning algorithm, and it often inserts many unintended (but logically correct) actions. For example, a planning algorithm might have the character Paul fall to the ground and roll around before standing up and walking to the table. Strictly speaking this is a logical solution to the problem. However, it is almost certainly not what the user intended.

[0005] In view of the above, there is a need for a method and system capable of inserting proper actions in an intelligent and method of meaningful way.

## **SUMMARY**

[0006] In accordance with an embodiment, there is provided a method for converting text for input into an animation generator. The method comprises: receiving said text; extracting at least one explicit action from said text; temporally organizing the at least one explicit action; associating an explicit goal for each one of the at least one explicit action; determining an un-achievability of the explicit goal based on an initial state associated with the at least one explicit action; upon the determining of the un-achievability, generating at least one implicit goal associated with an implicit action, the implicit action changing the initial state when executed, to render the explicit goal achievable; temporally organizing the explicit goal and the at least one implicit goal in a goal temporal organization; transmitting the goal temporal organization to the animation generator to generate an animation based on the goal temporal organization; and displaying the animation to represent the at least one explicit action associated with the explicit goal and the implicit action associated with the at least one implicit goal in time.

[0007] In accordance with another embodiment, there is provided a conversion engine for converting text for input into an animation generator. The conversion engine comprises: a goal generator for generating an explicit goal for each one of at least one explicit action in the text; and an implicit action planner in operative communication with the goal generator, the implicit action planner for: receiving the explicit goal associated to each one of the at least one explicit action comprised in the text; determining from a temporal organization of the at least one explicit action, an un-achievability of the explicit goal based on a initial state associated with the at least one explicit action; upon determining the un-achievability, generating at least one implicit action, the at least one implicit action rendering the explicit goal achievable by changing the initial state once executed; and transmitting the at least one implicit action to the goal generator; wherein the goal generator is adapted to receive the at least one implicit action; generate at least one implicit goal therefrom; and organize the at least one implicit goal with the explicit goal according to time in a

goal temporal organization, the goal temporal organization comprising instructions for the animation generator to generate an animation representative of the at least one explicit action and the at least one implicit action.

**[0008]** In accordance with yet another embodiment, there is provided a system for generating an animation from a text comprising at least one explicit action. The system comprises: a processor and a memory, the memory storing instructions for implementing the processor to: associate an explicit goal for each one of the at least one explicit action in the text; determine an un-achievability of the explicit goal based on an initial state; upon determining the un-achievability, generate at least one implicit goal associated with an implicit action, the implicit action changing the initial state when executed, to render the explicit goal achievable; temporally organize the explicit goal and the at least one implicit goal together in a goal temporal organization; and generate the animation based on the goal temporal organization, the animation representing the at least one explicit action associated with the explicit goal and the implicit action associated with the at least one implicit goal.

**[0009]** Throughout the present disclosure, the following terms are intended to refer to their meaning as provided below:

**[0010]** In the context of linguistics, a "predicate" is understood to be a feature of language that can be used to make a statement about something in the animation world. A predicate is an "animation word", which means that a predicate has a meaning understood by the animation generator. For example, the predicate "table" is understood by the animation generator that will display a graphical representation of a table. The predicate "on" associated to the predicate "table" is understood by associated to the animation generator as being indicative of a location. The animation generator understands that something will be positioned on the graphical representation of the table. An "action predicate" is understood to be a special kind of predicate, in the context of linguistics, that comes with "actant" slots and that refers specifically to an action concept.

**[0011]** "Actants" are the parameters (variables) of the predicates. Formally,  $give(X, Y, Z)$  is an example of an action predicate, with three "actants": (X, Y and Z).

We sometimes refer to the action predicate *give*, even though *give* is the name of the action predicate *give*(*X*, *Y*, *Z*).

**[0012]** A “fluent” refers to a logical predicate. The truth value of a fluent can change over time. The “arguments” of a fluent are elements of the world, such as actors, objects and the like.

**[0013]** An “actor” refers to a participant in an action. An “actor” can be active or passive, and animate or inanimate, for example.

**[0014]** A “role” refers to the part that an actor plays in the action. In the following example: “Peter and Paul walk to the table”, Peter, Paul and the table are actors, while the role of Peter and Paul is “agent”, and the role of the table is “destination”.

**[0015]** Furthermore, a “fluent” can be either primitive or defined. A “primitive fluent” is a fluent of which the value should be determined in order to determine the current state of the world. A “primitive fluent” is a fluent according to which parameters can vary within a same scene or from one scene to another. As a result, the value associated with a primitive fluent is context-specific. A “defined fluent” is a fluent that is defined with respect to other predicates. Defined fluents can be seen as shortcuts used in the precondition definition of actions. A defined fluent can be expressed in any predicate logic formulation, such as in a disjunctive normal form (DNF). For example, the fluent *canHold*(*X*,*Y*) can be expressed in DNF as:

$$\text{canHold}(X,Y) = (\text{oneFreehand}(X) \wedge \text{holdable\_with\_onehand}(Y)) \vee (\text{twoFreehands}(X) \wedge \text{holdable}(Y))$$
; where  $\wedge$  refers to the operator AND

**[0016]** The fluent *canHold*(*X*, *Y*) is true when the agent *X* can hold the object *Y* given that the object can be held and the agent has the number of free hands available to hold it. It should be understood that any predicate logic formula can be used.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

**[0017]** Further features and advantages of the present disclosure will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

[0018] Fig. 1 is a schematic illustration of a text-to-animation system with a conversion engine, in accordance with an embodiment;

[0019] Fig. 2 illustrates a temporal organization of explicit actions in the form of a graph, in accordance with an embodiment;

[0020] Fig. 3 is a flow chart illustrating a method for converting text for input into an animation generator, in accordance with an embodiment;

[0021] Fig. 4 illustrates a temporal organization of goals in the form of a goal graph, in accordance with an embodiment;

[0022] Fig. 5 is a block diagram of a conversion engine for converting explicit actions in a text into a temporal organization of both implicit and explicit goals, in accordance with an embodiment; and

[0023] Fig. 6 illustrates an example of an ontology organization, in accordance with an embodiment.

#### **DETAILED DESCRIPTION**

[0024] Figure 1 illustrates one embodiment of a TTM or TTA system 10. The system comprises a natural language processing (NLP) engine 12, a conversion engine 14 and an animation engine 16. The NLP module 12 receives a text 18 (also referred to as a script) inputted by a user of the system 10. The system 10 is used to convert the text 18 into an animation, video or movie 20. The system 10 also optionally has a display device (not shown) which receives and displays the animation thereon.

[0025] The NLP module 12 converts the input text 18 written in a natural language into a temporal graph 22, which is a graphical representation of explicit actions contained in the text 18. In the temporal graph 22, the actions are represented by nodes which are temporally organized. For example, the user inputs the following text: "Paul sits on a chair. Paul walks to the table".

[0026] As an example, Figure 2 illustrates one embodiment of a temporal graph 50 outputted by the NLP module 12. The temporal graph 50 is a graphical interconnection of nodes 52, 54, in which the nodes 52, 54, are temporally organized

(i.e. each are associated to a given point in time). In the temporal graph 50, each node 52 and 54 corresponds to an explicit action described in the text 18.

**[0027]** In one embodiment, the temporal graph 50 contains one Semantically Annotated Action Template (SAAT) at each node. These are illustrated as examples in Figure 2. A description of a SAAT can be found in a Patent Application entitled "TIME-ORDERED TEMPLATES FOR TEXT-TO-ANIMATION SYSTEM", with publication No. WO2008148211. Briefly, a SAAT is a template regrouping all information about the action predicate of the node. Semantically annotated description templates (SADT) are connected to each SAAT. A SADT regroups all information about one actant being in relation with the action predicate to which the SADT is connected. The SADTs are connected to a SAAT by arcs which express the semantic relation, or role, of the actants relative to their action predicate.

**[0028]** Referring back to Figure 2, a first SAAT corresponding to the action predicate "sit" occupies the first node 52 of the temporal graph 50. A SADT 56 representing Paul and a SADT 58 associated with "chair" are connected to the SAAT at node 52. The arcs 64 and 66 connecting the SADTs 56, 58 to the SAAT at node 52 describe the role of the actants in the SADTs. "Paul" is an agent (arc 64) of the action predicate, while "chair" acts as a location (arc 66) with respect to the action predicate. The SAAT at node 54 is associated with the action predicate "walk", and is connected to the SAAT at node 52 by an arrow 68 expressing that the action represented by the SAAT at node 54 occurs after the action associated with the SAAT at node 52. The actants "Paul" and "table" represented by the SADTs 60 and 62, respectively, are connected to the SAAT at node 54, by arc 64' and 66' respectively.

**[0029]** Now referring back to Figure 1, the conversion engine 14 receives the temporal graph 22 as an input and outputs a goal graph 24. While the temporal graph 22 is a temporal representation of the explicit action contained in the text 18, the goal graph 24 is an interconnection of goals which takes into account implicit actions which are not explicitly expressed in the text 18. Referring back to the example shown by the temporal graph of Figure 2, Paul cannot walk to the table from a sitting position: Paul inevitable has to stand up before proceeding to walk. Therefore, the

goal graph 24 outputted by the conversion engine 14 comprises an implicit goal corresponding to the implicit action "standing-up".

[0030] The conversion engine 14 receives the temporal graph 22 from the NLP module 12, and converts the temporal graph 22 into a goal graph 24. Each action located at a node in the temporal graph 22 is converted into a corresponding goal in the goal graph. A goal is said to be achieved when its corresponding explicit action can be executed. For each node of the goal graph 24, the conversion engine 14 determines if the goal can be achieved. The conversion engine 14 defines an initial state and a world theory associated with the particular goal. The initial state is specific to the particular goal, while the world theory is the same for all goals comprised in the analysed scene. The initial state is the initial state description prior to the execution of the action associated with the goal. The world theory can be seen as a description of the world. The world theory comprises a world view and an action theory. The world view regroups the data used to describe the possible states of the world. The action theory can be considered as a description of the actions predicates supported by the TTM or TTA system 10, including the preconditions to execute these actions and the effects of the actions.

[0031] In one embodiment, two types of logic predicates are used to describe the state of the world, namely rigid and flexible predicates. A rigid predicate refers to a predicate which has a fixed value for all states of a scene. A flexible predicate is a predicate of which the value varies from one state to another. A flexible predicate is also called a fluent. An example of a rigid predicate is a human (Paul). Paul is human independently of the state of the scene. Standing (Paul) is an example of a flexible predicate or fluent. This predicate may be true for some states of a scene, but false for other states of the same scene, such as a state in which Paul is sitting.

[0032] An action predicate is defined by various attributes such as *Implicit*, *Signature*, *Precondition*, *Post condition*, *State References*, *Typing*, and the like.

[0033] The *Implicit* attribute refers to a true/false value that determines whether the action associated with the action predicate can be included as an implicit action in a plan. The attribute *Implicit* increases the efficiency of the conversion performed by



the conversion engine 14, by restricting the possible actions to those that are most beneficial.

**[0034]** The *Signature* attribute provides the name of the action predicate and also provides the list of possible input parameters such as the agent or the location participating in the action. A *Signature* can be expressed as: sit (Agent, Location).

**[0035]** The *Precondition* attribute establishes a set of conditions that should be met in order for the action to be executed. In one embodiment, a precondition is expressed is a DNF formula. For example, the precondition DNF formula for the walk action is the following:

$$\text{not equal (Agent, Destination)} \wedge \text{standing (Agent)} \wedge \text{near (Agent, Origin)}$$

**[0036]** In this example, for the action to be executable, the agent cannot be the destination. The second expression of the DNF formula expresses that the agent has to be standing so that the action "walk" can be executed. Finally, the last part of the DNF expression indicates that the agent should be near the origin.

**[0037]** The *Post condition* attribute describes the effects of the action in the given state. This characteristic provides primitive fluents which are fluents that become true after the execution of the action, negated primitive fluent which are fluents that become false after the execution of the action, and conditional effects such as implications.

**[0038]** The *State References* attribute is used to simplify post conditions considered as overly complicated. A *State References* attribute refers to objects that are not included in the *signature*, such as free variables. When there is a free variable in an effect of the post condition, and this variable appears in a state reference, the effect applies to all the objects in the world that satisfies the state reference.

**[0039]** Taking the example of a "take" action, the *Signature* can be expressed as: holding (Agent, Object). The *state reference* can be expressed as: on (Object, F\_location), which means that the object is in F\_Location before the action occurs. A *Post condition* can be: holding(Agent, Object)  $\wedge$  not on(Object, F\_Location), which means that the object cannot be in F\_Location after the action occurred.

[0040] The *Typing* attribute refers to a list of rigid predicates used to specify a type or property for each parameter in the *Signature* attribute. An action can be added to a plan if these predicates are satisfied before doing the action. Taking the example of a walk predicate, a *Typing* attribute can be: human(Agent), location(Origin), location(Destination). The *Typing* attribute indicates that the agent should be human, and both the origin and the destination of the walk should be locations.

[0041] Finally, a *State Constraint* attribute refers to general constraints that apply to the state as a whole and act on all action predicates. A *State Constraint* allows a simplification of the post-conditions as it eliminates the need to include a given restriction in all the relevant action predicate definitions. For example, a *State Constraint* attribute can be: standing(X)  $\rightarrow$  not lying(X)  $\wedge$  not sitting(X). This particular *State Constraint* attribute specifies that when X is standing, then X cannot be lying nor sitting.

[0042] Now referring to Figure 3, there is described a method 100 for converting text for input into an animation generator. The text is descriptive of a scene in the animation.

[0043] In step 102, the text is received, as entered in a processing device by a user via a user interface for example.

[0044] In step 104, the explicit actions in the text are extracted therefrom, while, in step 106, the explicit actions are temporally organized according to a timeline of the animation. Step 104 optionally involves generating a temporal organization of nodes, each node representing an explicit action. The nodes are interconnected with one another in terms of a timing of the respective actions in the animation. The timing of each action is based on a textual timeline as described in the text. The temporal organization can take the form of a graph as illustrated in Figure 2.

[0045] It is noted that steps 102 and 104 are optionally performed in a natural language processing device. Step 106 may also be performed in such a language processing device. In this case, an additional step is performed to send the explicit actions and the temporal organization of these actions from the natural language

processing device to another processing device performing the remaining steps in the method 100.

**[0046]** In step 108, an explicit goal is associated to each explicit action or node in the temporal organization received or generated in step 106. Each of the explicit goals are then analyzed in step 110, to determine whether or not they are achievable or not (i.e. determine achievable and un-achievable explicit goals in the temporal organization). Step 108 is performed based on an initial state associated with the explicit action under analysis. The initial state provides a state of the scene before the explicit action is to be executed.

**[0047]** In step 112, when an un-achievable explicit goal is determined in step 110, at least one implicit goal associated with an implicit action is generated. The implicit action, once executed, changes the initial state associated with the explicit action, to render the explicit goal achievable. For example, an implicit action which is not found in the text, but which is to be executed before being able to execute the explicit action associated with the explicit goal under analysis, is determined, and an implicit goal associated with this implicit action is generated.

**[0048]** In step 114, the explicit goals and the generated implicit goals are temporally organized in a goal temporal organization. In one embodiment, the generated goals are inserted in the temporal organization formed in step 106 (or alternatively received from a natural language processing device), before the analyzed explicit goal. The goal temporal organization can also take the form of a goal graph with interconnections of nodes corresponding to either implicit or explicit goals to be executed with respect to time.

**[0049]** In step 116, the final goal temporal organization of step 114 is outputted and transmitted to an animation generator to generate an animation, movie or sequence of frames.

**[0050]** In step 118, the animation is optionally displayed to represent the explicit actions associated with the explicit goals, and the implicit actions associated with the implicit goals according to time.

[0051] Still referring to Figure 3, in one embodiment, either one or both of steps 106 and 114 involve generating an action and a description template for each node in the interconnection of nodes. The action template is filled with action predicate information associated with the action of the node. The description template is filled with information relative to an actant associated with the action predicate for example. The action template and the description template are semantically related to one another as described above in relation to Figure 2 for example.

[0052] In one embodiment, step 110 involves defining a world theory associated with the explicit goal, the world theory being used in step 112 to generate the implicit goal and related implicit action.

[0053] The above described method 100 can be embodied in any processing device which is implemented according to instructions stored in a memory, where the instructions allow the above method to be achieved. Such a processing device and memory can take the form of a computer for example, or a server accessible to users or other applications, via a network connection.

[0054] Referring now to Figure 4 and the example input text "Paul sits on a chair. Paul walks to the table", the conversion engine 14 generates a goal graph 70 in which the first node 72 is associated with the first SAAT at node 52 of the temporal graph 50 in Figure 2, namely "Paul sits on a chair". The second node 74 of the goal graph 70 in Figure 4 is associated with the second SAAT at node 54 of the graph 50 of Figure 2, namely "Paul walks to the *door*". In this particular case, the world theory consists of a description of the world around Paul, the actions that Paul is allowed to make, as well as all the rules that govern the actions and the surroundings of Paul. Taking the example of the second goal, "*Paul walks to the table*", the initial state corresponds to Paul sitting on the chair, and the analyzed goal corresponds to Paul walking to the table. Using the world theory, the conversion engine 14 determines that the analyzed goal cannot be achieved from the initial sitting position since Paul has to be in a standing position before being able to walk. The action "stand up" is missing in order to execute the action "*walk*". The conversion engine 14 then inserts a node 76 corresponding to the missing action between the nodes 72 and 74. The

goal 1' associated with the "stand up" action is placed at the node 76. This goal is associated to an implicit action which was not explicitly found in the input text.

**[0055]** Now in reference to Figure 5, in one embodiment, the conversion engine 14 (also referred to as a conversion module) comprises a goal graph generator (GGG) 80 and a planner 82. The goal planner 82 comprises a converter 84 in communication with an implicit action planner 86 and a memory 88. The GGG 80 receives the temporal graph 22 from the NLP module 12 (refer to Figure 1). The temporal graph 22 comprises time organized SAATs, each SAAT presenting an action predicate accompanied with all of its conditions, and the actors that participate in the SAAT with their associated role. Each action predicate condition is a logic formula over fluents of various types which take actors as arguments.

**[0056]** The GGG 80 generates a goal for each SAAT node of the temporal graph 22. The generated goals corresponds to explicit actions comprised in the text 18 inputted by the user. For each generated *goal*, the GGG 80 generates a Planner Request Context (PRC) object to be used by the planner in order to determine whether any implicit action should be executed prior to the execution of the generated goal. A PRC object comprises the contextual information used by the planner 82 to generate its plan. The contextual information comprises the list of actors in the scene and their associated conceptual assets, and the list of pre-conditions of the action predicate. A conceptual asset is a predicate which takes part in an action, such as nouns. The predicates "table", "car", "box", "human", and "hand" are examples of conceptual assets.

**[0057]** The converter 84 of the goal planner 82 receives the PRC request and converts it into a logic programming language. It should be understood that the PRC request can be converted into any logic programming language known to a person skilled in the art, to form a logically programmed request. The request converted in logic programming language is then sent to the implicit action planner 86 which determines whether an implicit goal should be inserted prior to the analyzed goal.

**[0058]** The implicit action planner 86 first receives an identification of the conceptual asset type of the actors associated with the goal.

[0059] In reference to Figure 6, each actor is associated with a conceptual asset coming from an ontology of concepts. An ontology organization 200 takes the form of an oriented graph in which nodes 202, 204 and 206 are connected together, as illustrated in Figure 6. Each lower node is a particular case of an upper node. As a result, the instance "Peter" at node 206 is a particular case of the concept "man" at node 204, which in turn is a particular case of the concept "human" at node 202. Each level of the ontology 200 has a given set of properties, such as holdable, movable, wearable, readable, etc., which govern the way it is treated by the planner 86. In one embodiment, the implicit action planner 86 supports a limited number of conceptual asset types. In this case the implicit action planner 86 associates each supported conceptual asset type. For example, if the concept "man" is not supported by the implicit action planner 86, the actor "Peter" is associated with the conceptual asset type "human" and the properties of the conceptual asset type "human" are associated with the actor "Peter". The ontology of concepts and the properties associated with each level of the ontology are stored in the memory 88.

[0060] Then the implicit action planner 86 expresses the conditions on the SAATs of the temporal graph 22 in terms of fluents which are stored in memory 88. In one embodiment, the memory stores two types of fluents, namely primitive fluent and defined fluents. Concerning primitive fluents, the implicit action planner 86 receives their value for each actor present in the scene. The implicit action planner 86 infers the value of the defined fluents from the values of the primitive fluents.

[0061] The implicit action planner 86 also asserts the logic programming terms (also referred to as a set of logically programmed terms) received from the generator 84 and corresponding to the fluents that are true in the initial state. Referring back to the example, the fluent Standing (Peter) is asserted as being true. The implicit action planner 86 asserts the logic programming terms that express the actors in the scene as being conceptual assets. For example, the fluent Human (Peter) is considered to be true. The implicit action planner 86 also asserts the logic programming terms that express the properties of the actors/ conceptual asset type previously determined, such as Movable (Peter) and it asserts the logic programming term that expresses the pre-conditions of the goal using the terms previously asserted. This term can be

expressed in clausal normal form (CNF) (please confirm) or any other logic programming form, such as in the following expression: [[Standing(Peter), Near(Peter, Table)], [Sitting (Peter), On(Peter, Chair)] ].

**[0062]** The previously asserted pre-condition represents a condition for executing the explicit action. The implicit action is then transmitted to the GGG 80 which generates an implicit goal according to this condition. This implicit goal is inserted above the analyzed goal in the goal graph 24. The GGG 80 finally transmits the goal graph to the animation engine 16 in order to generate the movie or animation 20.

**[0063]** While the present description refers to a single implicit action and a single corresponding implicit goal, it should be understood that more than one implicit goal may be inserted so that an analyzed goal be achieved. Any planning algorithm known to a person skilled in the art may be used by the GGG to generate the goal graph. The NLP engine 12 and the conversion engine 14 can be integrated in a single engine receiving a text as input and outputting a goal graph comprising implicit actions.

**[0064]** While preferred embodiments have been described above and illustrated in the accompanying drawings, it will be evident to those skilled in the art that modifications may be made therein without departing from the essence of this disclosure. Such modifications are considered as possible variants comprised in the scope of the disclosure.

**CLAIMS:**

1. A method for converting text for input into an animation generator, the method comprising:

receiving said text;

extracting at least one explicit action from said text;

temporally organizing the at least one explicit action;

associating an explicit goal for each one of the at least one explicit action;

determining an un-achievability of the explicit goal based on an initial state associated with the at least one explicit action;

upon the determining of the un-achievability, generating at least one implicit goal associated with an implicit action, the implicit action changing the initial state when executed, to render the explicit goal achievable;

temporally organizing the explicit goal and the at least one implicit goal in a goal temporal organization;

transmitting the goal temporal organization to the animation generator to generate an animation based on the goal temporal organization; and

displaying the animation to represent the at least one explicit action associated with the explicit goal and the implicit action associated with the at least one implicit goal in time.

2. The method of claim 1, wherein the receiving the text comprises receiving the text at a natural language processing engine; and wherein at least one of the extracting and the temporally organizing the at least one explicit action are performed at the natural language processing engine.

3. The method of claim 1, wherein the temporally organizing the at least one explicit action comprises generating an interconnection of nodes, each one of the



nodes corresponding to each one of the at least one explicit action and being associated to a point in time in the animation to be generated.

4. The method of claim 3, wherein the temporally organizing the explicit goal and the at least one implicit goal comprises inserting a node in the interconnection of nodes, the node corresponding to the at least one implicit goal.

5. The method of claim 3, wherein the generating the interconnection of nodes comprises generating an action template for each one of the nodes, the action template comprising action predicate information corresponding to the at least one explicit action.

6. The method of claim 5, wherein the generating the interconnection of nodes comprises:

generating a description template for each one of the nodes, the description template comprising information on an actant associated with the action predicate.

7. The method of claim 6, comprising connecting the action template to the description template based on a semantic relation between the actant and the action predicate.

8. The method of claim 1, wherein the determining of the un-achievability of the explicit goal comprises defining a world theory associated with the explicit goal, the world theory comprising at least one of possible states of a scene of the animation in which the at least one explicit action takes place, and at least one action predicate achievable in the scene.

9. The method of claim 8, wherein the defining of the world theory comprises defining the at least one of the possible states from at least one logic predicate, the at least one logic predicate being defined by at least one of: a fixed value applicable to all of the possible states; and a variable value applicable to a particular one of the possible states.

10. The method of claim 9, wherein the defining the world theory comprises providing an attribute to the at least one action predicate achievable in the scene, the attribute defining at least one condition to be met for an action corresponding to the at least one action predicate to be achievable.

11. The method of claim 10, wherein the generating the at least one implicit goal comprises generating the at least one implicit goal based on the at least one of the possible states and the at least one condition associated with the at least one action predicate achievable.

12. A conversion engine for converting text for input into an animation generator, the conversion engine comprising:

- a goal generator for generating an explicit goal for each one of at least one explicit action in the text; and

- an implicit action planner in operative communication with the goal generator, the implicit action planner for: receiving the explicit goal associated to each one of the at least one explicit action comprised in the text; determining from a temporal organization of the at least one explicit action, an un-achievability of the explicit goal based on a initial state associated with the at least one explicit action; upon determining the un-achievability, generating at least one implicit action, the at least one implicit action rendering the explicit goal achievable by changing the initial state once executed; and transmitting the at least one implicit action to the goal generator;

- wherein the goal generator is adapted to receive the at least one implicit action; generate at least one implicit goal therefrom; and organize the at least one implicit

goal with the explicit goal according to time in a goal temporal organization, the goal temporal organization comprising instructions for the animation generator to generate an animation representative of the at least one explicit action and the at least one implicit action.

13. The conversion engine of claim 12, wherein the goal generator is adapted to generate a planner request context (PRC) object comprising the explicit goal and contextual information of a scene of the animation, for the implicit action planner to determine the un-achievability of the explicit goal based on the PRC.

14. The conversion engine of claim 13, wherein the goal generator is adapted to provide, as the contextual information of the PRC object: an identification of an actor with at least one associated conceptual asset, and at least one condition associated with at least one action predicate.

15. The conversion engine of claim 13, comprising a converter in operative communication with the goal generator and the implicit action planner, for converting the PRC object into a logically programmed request, and relaying the logically programmed request to the implicit action planner.

16. The conversion engine of claim 15, wherein the implicit action planner is adapted to determine the un-achievability from the logically programmed request and a given conceptual asset type associated with the contextual information in the logically programmed request.

17. The conversion engine of claim 13, wherein the at least one associated conceptual asset is based on an ontology organization defining at least one link between the at least one associated conceptual asset and other conceptual assets.

18. The conversion engine of claim 16, wherein the implicit action planner is adapted to assert a set of logically programmed terms of the logically programmed request to provide a condition under which the at least one explicit action associated with the explicit goal is achievable.

19. The conversion engine of claim 12, wherein the implicit action planner is adapted to generate the at least one implicit action based on the condition.

20. A system for generating an animation from a text comprising at least one explicit action, the system comprising:

a processor; and

a memory for storing instructions for implementing the processor to:

associate an explicit goal for each one of the at least one explicit action in the text;

determine an un-achievability of the explicit goal based on an initial state;

upon determining the un-achievability, generate at least one implicit goal associated with an implicit action, the implicit action changing the initial state when executed, to render the explicit goal achievable;

temporally organize the explicit goal and the at least one implicit goal together in a goal temporal organization; and

generate the animation based on the goal temporal organization, the animation representing the at least one explicit action associated with the explicit goal and the implicit action associated with the at least one implicit goal.

21. The system of claim 20, comprising a natural language processing engine implemented to:

receive the text;

extract at least one explicit action from the text; and

temporally organize the at least one explicit action.

22. The system of claim 20, comprising a display device for displaying the animation.

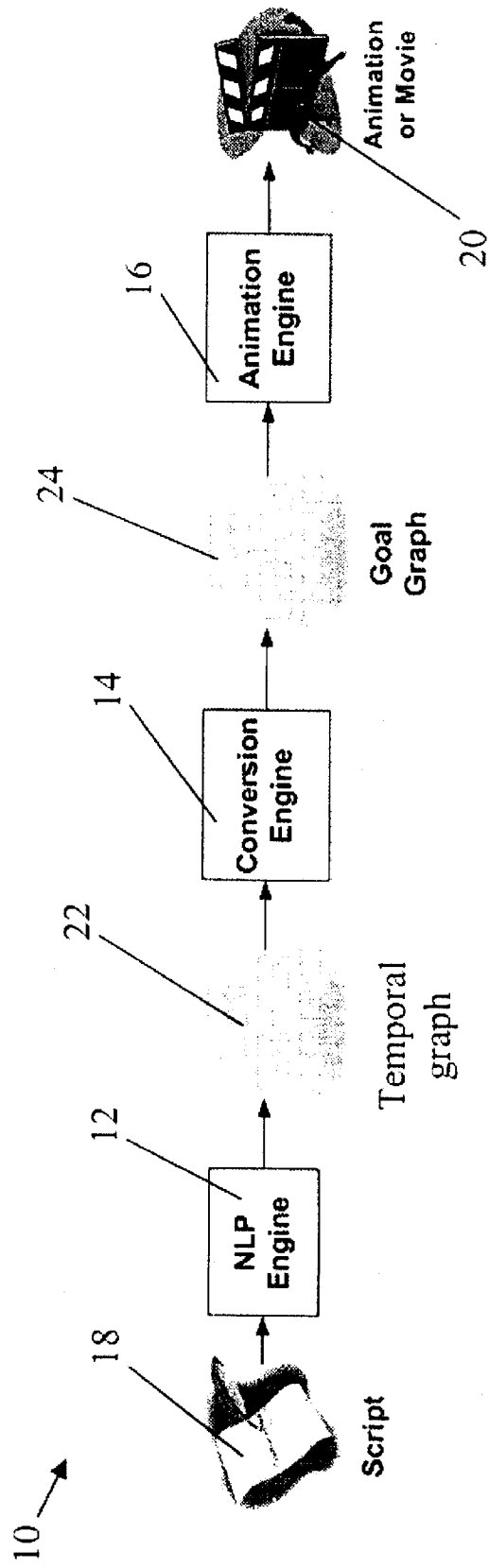


Fig. 1

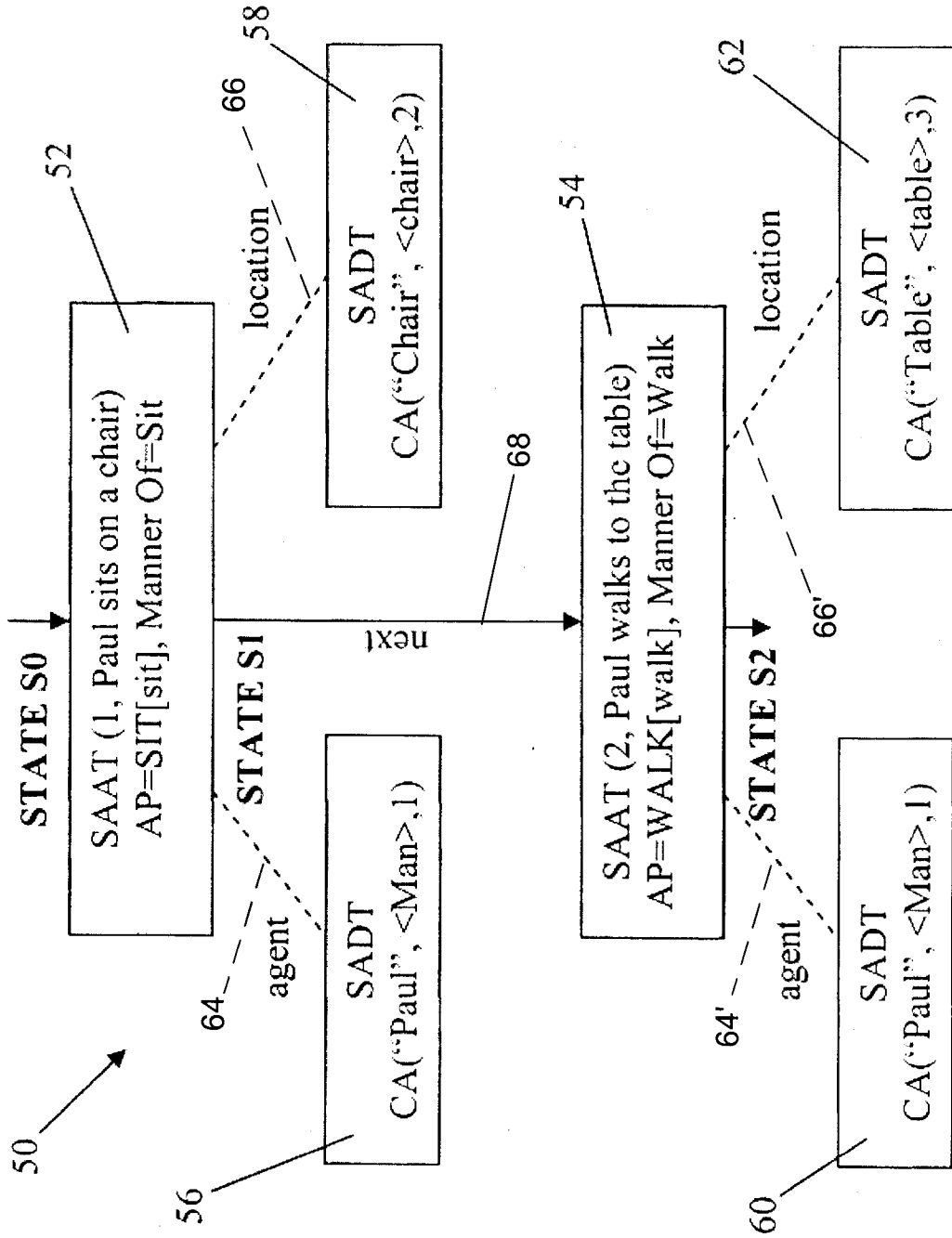


Fig. 2

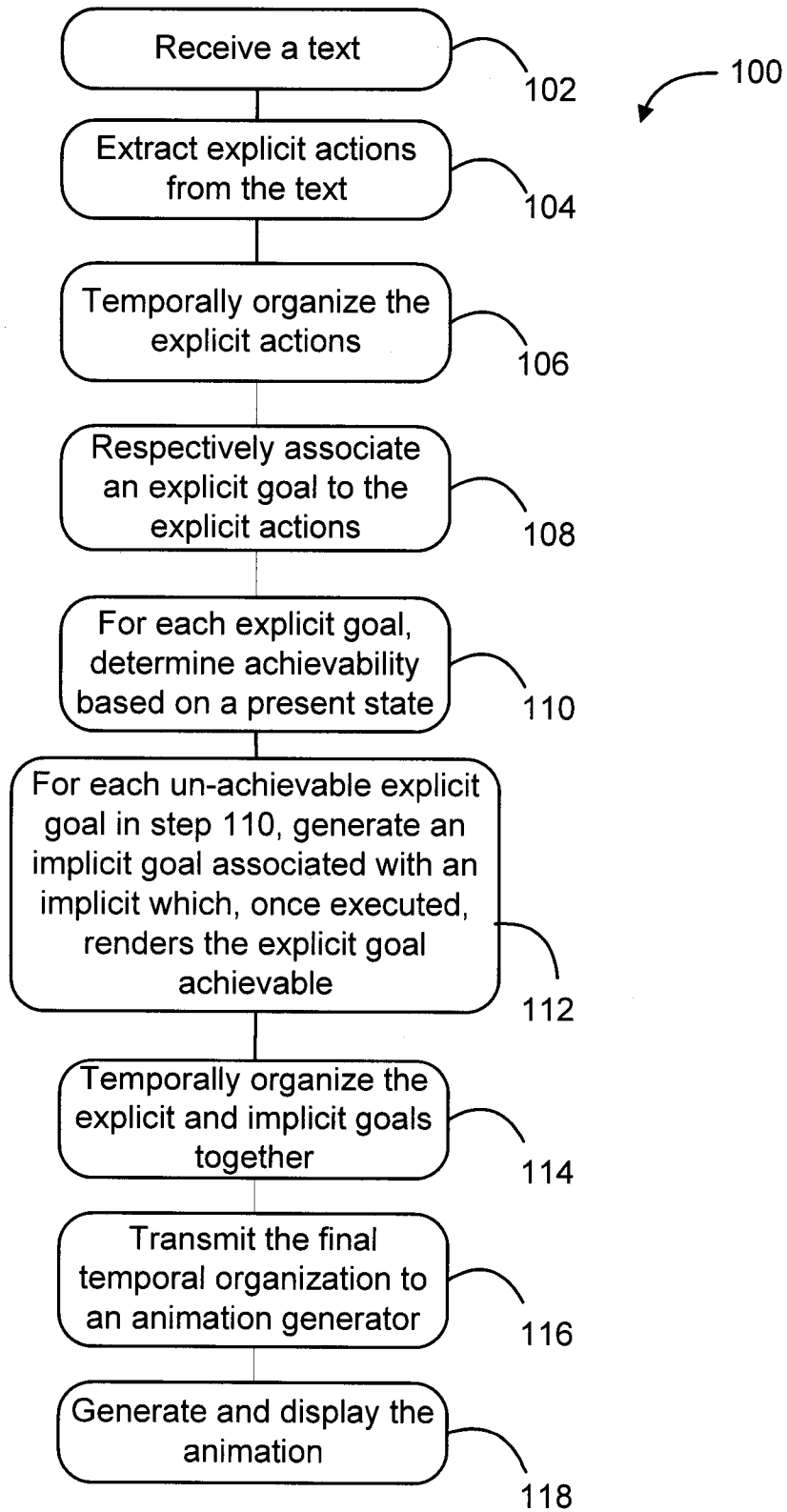


Fig. 3



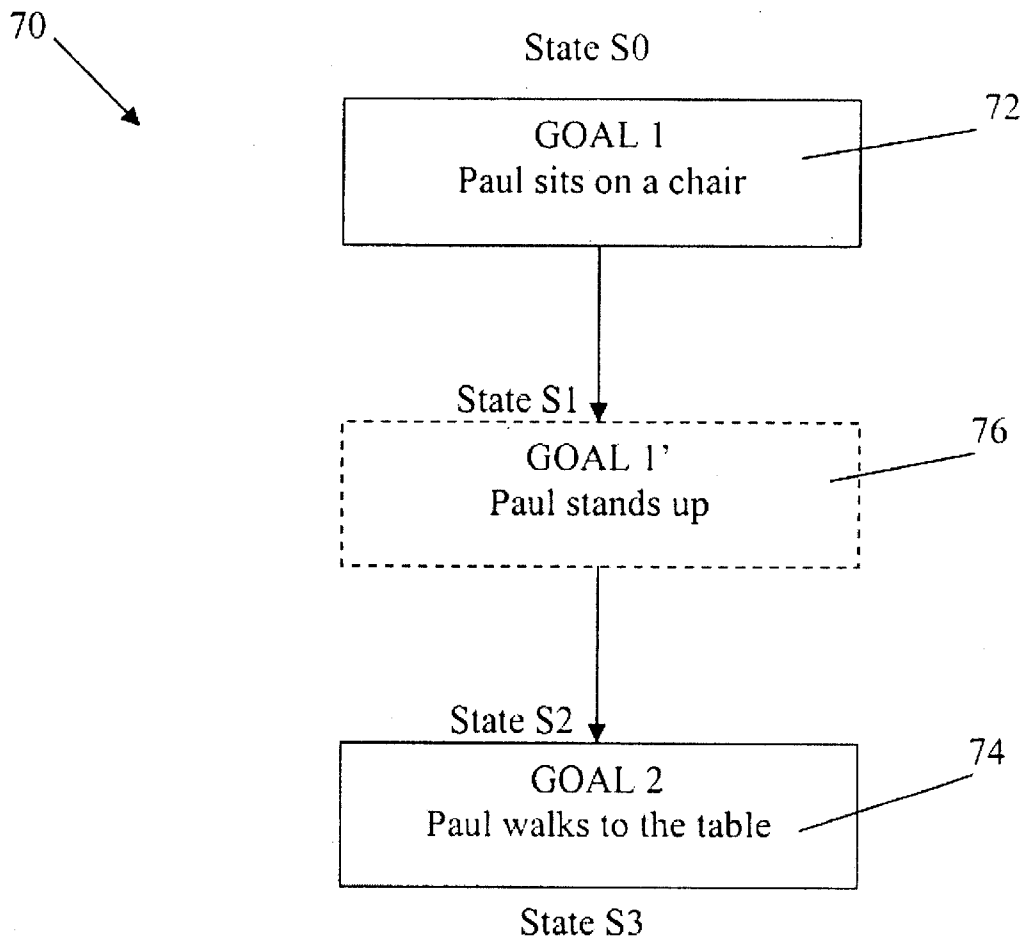


Fig. 4

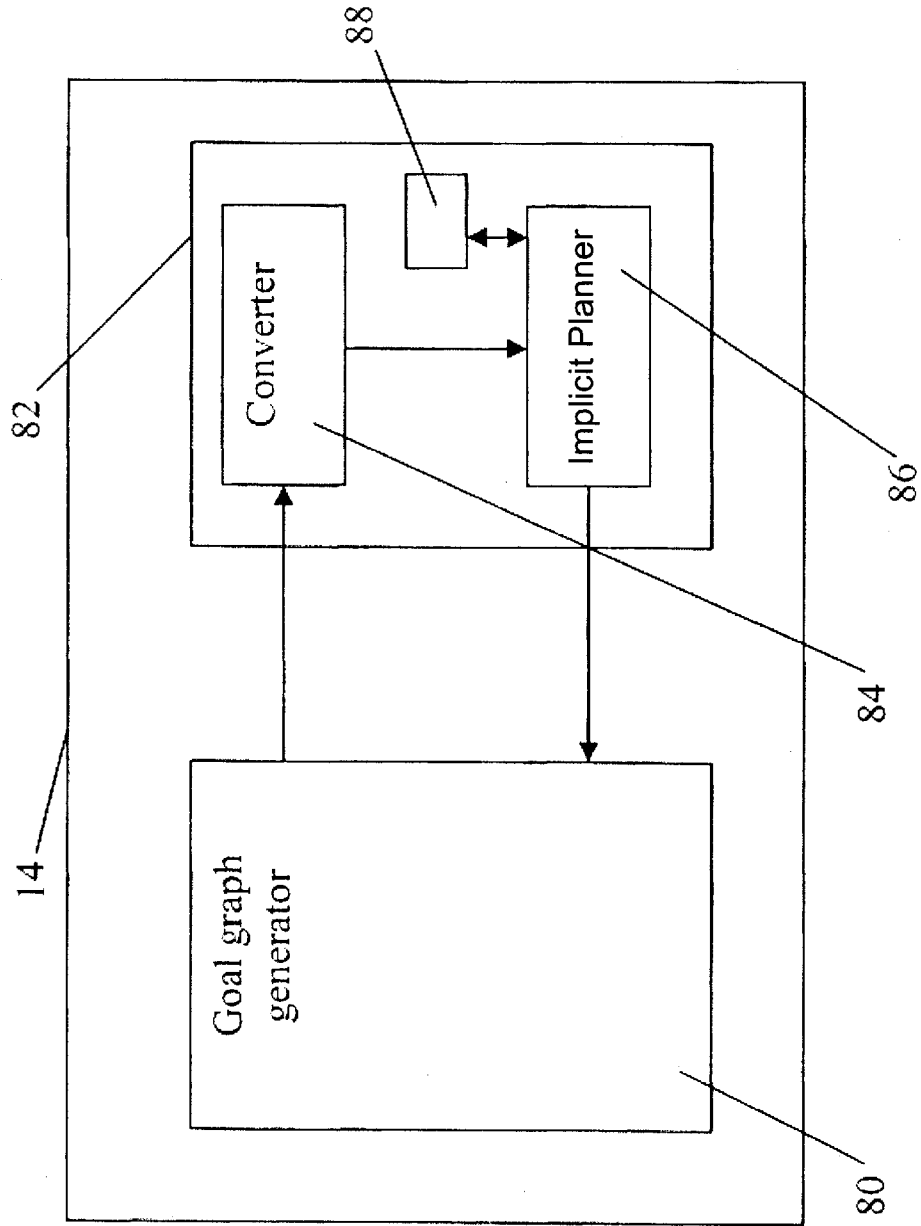


Fig. 5

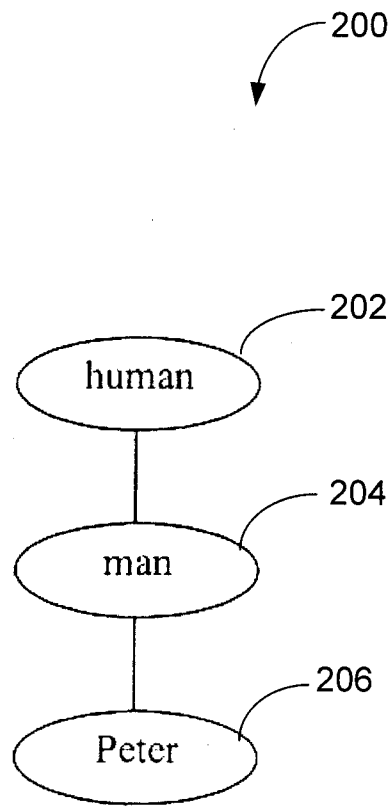


Fig. 6

**INTERNATIONAL SEARCH REPORT**

International application No.  
PCT/CA2009/001518

| <p>A. CLASSIFICATION OF SUBJECT MATTER<br/>                 IPC: <i>G06F 17/20</i> (2006.01) , <i>G06F 17/27</i> (2006.01) , <i>G06T 13/00</i> (2006.01) , <i>G06T 15/70</i> (2006.01)<br/>                 According to International Patent Classification (IPC) or to both national classification and IPC</p>   |   |  |  |   |                       |   |  |      |   |  |      |
|---|---|--|--|---|-----------------------|---|--|------|---|--|------|
| <p>B. FIELDS SEARCHED</p> <p>Minimum documentation searched (classification system followed by classification symbols)<br/>                 IPC: <i>G06F 17/20</i> (2006.01) , <i>G06F 17/27</i> (2006.01) , <i>G06T 13/00</i> (2006.01) , <i>G06T 15/70</i> (2006.01)</p> <p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields</p> <p>Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms<br/>                 Epoque, CPD, Qweb (using keywords): animate, animation, animation generator, animation generation, text-to-animation (TTA), text-to-movie (TTM), goal, natural language, natural language processing, temporal, implicit, explicit</p>  |   |  |  |   |                       |   |  |      |   |  |      |
| <p>C. DOCUMENTS CONSIDERED TO BE RELEVANT</p> <table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width:10%;">Category</th> <th style="width:60%;">Citation of document, with indication, where appropriate, of the relevant</th> <th style="width:30%;">Relevant to claim No.</th> </tr> </thead> <tbody> <tr> <td align="center">A</td> <td>US 2006 0197764 A1 (YANG) 7 September 2006 (07-09-2006)<br/>Entire document</td> <td align="center">1-22</td> </tr> <tr> <td align="center">A</td> <td>US 7,016,828 B1 (COYNE ET AL.) 21 March 2006 (21-03-2006)<br/>column 14, line 42 - column 15, line 16</td> <td align="center">1-22</td> </tr> </tbody> </table>  |   |  | Category   | Citation of document, with indication, where appropriate, of the relevant   | Relevant to claim No. | A | US 2006 0197764 A1 (YANG) 7 September 2006 (07-09-2006)<br>Entire document | 1-22 | A | US 7,016,828 B1 (COYNE ET AL.) 21 March 2006 (21-03-2006)<br>column 14, line 42 - column 15, line 16 | 1-22 |
| Category  | Citation of document, with indication, where appropriate, of the relevant   | Relevant to claim No.  |  |   |                       |   |  |      |   |  |      |
| A   | US 2006 0197764 A1 (YANG) 7 September 2006 (07-09-2006)<br>Entire document  | 1-22   |  |   |                       |   |  |      |   |  |      |
| A   | US 7,016,828 B1 (COYNE ET AL.) 21 March 2006 (21-03-2006)<br>column 14, line 42 - column 15, line 16  | 1-22   |  |   |                       |   |  |      |   |  |      |
| <p><input type="checkbox"/> Further documents are listed in the continuation of Box      <input checked="" type="checkbox"/> See patent family annex.</p> <table border="1" style="width:100%; border-collapse: collapse;"> <tbody> <tr> <td style="width:50%; vertical-align: top;"> <p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </td> <td style="width:50%; vertical-align: top;"> <p>"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</p> <p>"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</p> <p>"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</p> <p>"&amp;" document member of the same patent family</p> </td> </tr> </tbody> </table> |   |  | <p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> | <p>"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</p> <p>"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</p> <p>"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</p> <p>"&amp;" document member of the same patent family</p> |                       |   |  |      |   |  |      |
| <p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>  | <p>"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</p> <p>"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</p> <p>"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</p> <p>"&amp;" document member of the same patent family</p> |  |  |   |                       |   |  |      |   |  |      |
| <p>Date of the actual completion of the international search<br/>17 December 2009 (17-12-2009)</p>  |   | <p>Date of mailing of the international search report<br/>13 January 2010 (13-01-2010)</p> |  |   |                       |   |  |      |   |  |      |
| <p>Name and mailing address of the ISA/CA<br/>                 Canadian Intellectual Property Office<br/>                 Place du Portage I, C114 - 1st Floor, Box PCT<br/>                 50 Victoria Street<br/>                 Gatineau, Quebec K1A 0C9<br/>                 Facsimile No.: 001-819-953-2476</p>  |   | <p>Authorized officer<br/>                 Kristy Hyam (819) 934-2673</p>                  |  |   |                       |   |  |      |   |  |      |

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.  
**PCT/CA2009/001518**

| Patent Document<br>Cited in Search Report | Publication<br>Date | Patent Family<br>Member(s) | Publication<br>Date |
|---|---------------------|----------------------------|---------------------|
| US 2006197764A1                           | 07-09-2006          | US 7412389B2               | 12-08-2008          |
| US 7016828B1                              | 21-03-2006          | US 7373292B1               | 13-05-2008          |