AI-based Animation for Interactive Storytelling

Marc Cavazza, Fred Charles and Steven J. Mead
School of Computing and Mathematics, University of Teesside
TS1 3BA Middlesbrough, United Kingdom
{m.o.cavazza, f.charles, steven.j.mead}@tees.ac.uk

Abstract

In this paper, we describe a method for implementing AI-based animation of artificial actors in the context of interactive storytelling. We have developed a fully implemented prototype based on the Unreal™ game engine and carried experiments with a simple sitcom-like scenario. We discuss the central role of artificial actors in interactive storytelling and how real-time generation of their behaviour participates to the creation of a dynamic storyline. We follow previous work describing the behaviour of artificial actors through AI planning formalisms, and adapt it to the context of narrative representation. The set of all possible behaviours, accounting for various instantiations of a basic plot, can be represented through an AND/OR graph. A real-time variant of the AO* algorithm can be used to interleave planning and action, thus allowing characters to interact between themselves and with the user. Finally, we present several examples of short plots and situations generated by the system from the dynamic interaction of artificial actors.

Keywords: Interactive Storytelling, AI-based Animation, Autonomous Characters, Virtual Humans

1. Introduction

The development of artificial actors and AI-based animation naturally leads to envision future interactive storytelling systems. A typical interactive storytelling system would be based on autonomous virtual actors that generate the plot through their real-time interaction. Besides the user should be allowed to interfere with the ongoing action, thereby altering the plot as it unfolds.

Many interactive storytelling models have been proposed: emergent storytelling [Dautenhahn, 1998], interactive virtual storytelling [Perlin and Goldberg, 1996] [Nakatsu and Tosa, 1999], user-centered plot resolution [Sgouros et al., 1996], character-driven storytelling [Mateas, 1997] [Young, 2000]. Previous work has identified relevant dimensions and key problems for the implementation of interactive storytelling, among which: the status of the user, the level of explicit narrative representation and narrative control, the modes of user intervention, the relations between characters and plot, etc.

Our own conception of interactive storytelling is strongly character-centered [Young, 2000]. As a consequence, it privileges anytime interaction and occasional involvement of the user. The long-term applications we envision are interactive stories, acted by artificial characters, which rely on an initially well-defined scenario, but that the user can alter by interfering at anytime with the ongoing action.

It appears that exploring actors' behaviour in storytelling is more feasible within narrative genres that display the simplest storylines. Developing "virtual sitcoms" seems a relevant first step in the pursuit of interactive storytelling. As its own name suggests, sitcom standing for "situation comedy", a significant fraction of the story interest arises from the situations into which the actors find themselves. This is thus an ideal testbed to reproduce the emergence of narratively meaningful situations from the combined AI-based animation of virtual actors.

Throughout this paper, we will illustrate the discussion with the first results obtained from a proof-of-concept “virtual sitcom” prototype. The prototype we describe has been developed using the Unreal™ game engine. The Unreal™ environment provides most of the user interaction features required to support user intervention in the plot, such as navigating in the environment and interacting with objects in the virtual set and its use has been previously reported in prototyping interactive storytelling [Young, 2000]. The system described in this paper has been fully implemented as a set of template C++ classes, and is used as a native function by UnrealScript™, Unreal™’s scripting language.

In the next sections, we discuss how characters’ behaviours can be defined from the properties of a story genre. We then describe the planning technology used to generate character’s behaviour and present various example of how a specific plot can emerge from characters’ interaction.

2. Character Behaviour and Plot Representation

Plans are the most generic description of an artificial actor’s behaviour, both in AI-based animation [Webber et al., 1994] [Geib and Webber, 1993] [Funge et al., 1999] and in character-centered interactive storytelling [Young, 2000] [Mateas, 1997]. Implementing AI planning for real-time behaviour of artificial actors is always a challenging task. Further, there are two specific requirements for planning in the context of interactive storytelling. One is the need for re-planning, i.e. producing a new plan following action failure or user intervention. The other is an authoring constraint: plans should be representations of the storyline through a character’s behaviour.

On the other hand, the formalisation of stories can be tracked back to Aristotle’s Poetics. In modern times, Propp [1968] has founded the formal description of narratives, through the notion of narrative functions. Structuralists like Barthes proposed structured (“stemma-like”) representation of narrative episodes [Barthes, 1966]. Schank later introduced scripts and plans as a computational version of the structuralist formalisation of stories [Schank and Abelson, 1977]. Paradoxically though, Schank’s approach was still very much a cognitive one, with no unified representation of specifically narrative knowledge.

As a narrative representation, a single plan, rather than being only a problem-solving procedure, is representing the set of potential plot instances. It can be seen as a generic formalism and as a resource for story generation. This is how the plan can represent the storyline through a character’s behaviour. While there are no straightforward rules to convert high-level narrative functions into characters’ plans, we have attempted at devising specific rules that could be applicable in the context of the simple genre (sitcoms) that we are experimenting with. The basic hypothesis is that the overall story will emerge from the relations that exist between the various actors’ plans, these relations being determined from the story genre. For instance, if Ross plan is to seduce Rachel and Rachel’s plan just consists in carrying on her daily activities unaware of Ross, this is likely to result in a series of comic misunderstandings. We have hence defined separate plans for Ross and Rachel, which are in agreement with properties of the sitcom genre. Ross plan is to invite Rachel out for dinner. This plan is decomposed into a first set of high-level sub-goals: acquiring information about Rachel, attracting her (positive) attention, finding a way to talk to her privately, etc. Each sub-goal can be subsequently refined into many different options that constitute elements of alternative plots (Figure 1). On the other hand, Rachel's plan is not specifically oriented towards Ross. Her plan will lead her to carry various activities, socially or privately, as a function of her mood and sociability.

In our current prototype, only the two principal characters (Rachel and Ross) have their behaviours governed by plans. We have created a set of other, secondary characters (e.g. Phoebe, Monica), whose behaviour is essentially reactive and determined by scripted rules. For instance, they would carry certain actions if asked by the main characters or their mood will change when interacting with them (e.g. Phoebe would be upset if Ross behaves rudely).

3. AI Planning Techniques and Behaviour Generation

As described previously, the first step consists in describing the overall characters' plan from the narrative content. We represent a character's plan using a Hierarchical Task Network, which is formalised as and AND/OR graph. As we are representing narrative content a priori, our representations are actually explicit graphs (and this has implications for their automatic processing). From a formal perspective, the search process that is carried out by an AI planner takes an AND/OR graph and generates from it an equivalent state-space graph [Nilsson, 1998]. The process by which a state-space graph is normally produced from a Hierarchical Task Network (HTN) is called *serialisation* [Tsuneto et al., 1997]. However, when the various sub-goals are independent from one another, the planner can build a solution straightforwardly by directly searching the AND/OR graph without the need for serialising it [Tsuneto et al., 1997].

In the case of storytelling, the sub-goals are independent as they represent various stages of the story[1]. Decomposability of the problem space derives from the inherent decomposition of the story into various stages or scenes, a classical representation for stories [Schank and Abelson, 1977]. The solution takes the form of a sub-graph (rather than a path like in traditional graph search). In our context, the terminal nodes of this sub-graph corresponds to a sequence of actions that constitute a specific instantiation of the storyline. The low-level (motion) animations rely on Unreal™'s built-in mechanisms. Yet, the formal solution cannot be reduced to a script containing this sequence of actions, as the hierarchical representation contains information of narrative relevance.

The use of graph search seems to have many representational advantages over other formalisms such as Finite-State Automata, which are frequently used as compiled plans [Geib, 1994] [Kurlander and Ling, 1995]. This helps maintaining a unified representation relating character's behaviour and personalities to potential storylines. Further, there has been recently a renewed interest in search-based planning techniques, as these have demonstrated significant performance on various planning tasks [Tsuneto et al., 1997] [Bonet and Geffner, 1999] [Korf, 1996] [Pemberton and Korf, 1994]. We use the AO* algorithm [Nilsson, 1980] [Pearl, 1984] [Knight and Rich, 1991] to search the AND/OR graph. The AO* algorithm is a heuristic search algorithm operating on AND/OR graphs: it can find an optimal solution sub-graph according to its evaluation functions. It can be described as comprising a top-down and a bottom-up component. The top-down step consists in expanding OR nodes, using a heuristic function, to find a solution basis, i.e. the most

---

[1] There is some level of long-range dependency, as some early actions may render future actions inapplicable. Even so, this mainly reduces the search space without affecting previous choices: in planning terms, the delete-list of planning operators remains empty.

promising sub-graph. For instance, in the tree of Figure 1, the “acquire information” node can be expanded into different sub-goals, such as “read Rachel’s diary” or “ask one of her friends”. The actual choice of sub-goal will depend on the heuristic value of each of these sub-goals, which contains narrative knowledge, such as the actor’s personality. However, what ultimately characterises a solution graph is not the cost of the edges that constitute it but rather the set of values attached to its terminal nodes. This is why the evaluation function of each previously expanded node has to be revised according to these terminal values. This is done using a rollback function [Pearl, 1984], which is a recursive weighting function that aggregates individual evaluation functions along successor nodes. In the context of interactive storytelling, this bottom-up step can be used to take into account an action’s outcome, when planning and action are interleaved (which is the case in our prototype).

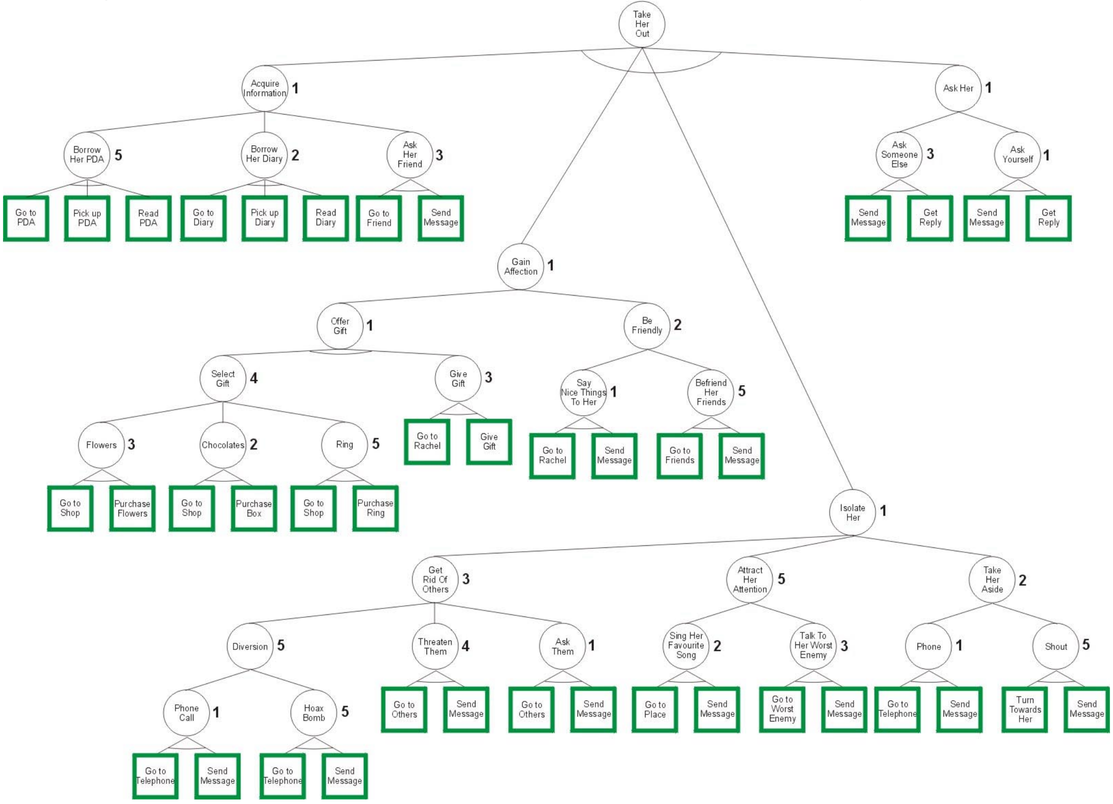


Figure 1-Ross' Plan

In interactive storytelling, several actors, or the user himself, might interfere with one agent’s plans, causing its planned actions to fail. Hence, the story can only carry forward if the agent has re-planning capabilities. Whenever an action fails, the heuristic value for the corresponding node is set to a “futility” value (i.e., equivalent to an “infinite cost” for that terminal node), and a new solution graph is computed. The new solution would take into account action failure by propagating its updated value to its parent nodes through the rollback mechanism. In any case, failed actions cannot be undone, as they have been played on stage. Action failure is indeed part of the story itself. This is why the dramatisation of actions must take their possible failure into account and store corresponding animations. The need to dramatise action failure can have implications also for transition to the next action undertaken, as proper conditions might in some case have to be restored (for instance, if the next action cannot be performed from the location where the previous action failed). This would mean that not only failure, but recovery from failure might need be dramatised as well. However, we have not encountered this problem in the scenarios we have described so far, which remain relatively simple.

Considering the need for anytime interaction, we have developed a “real-time” variant of AO* that does not compute a complete solution sub-graph but interleaves planning and execution and only computes the partial solution tree required to carry out the next action. It explores the tree in a depth-first, left-to-right fashion [Pemberton and Korf, 1994] using essentially the heuristic part of evaluation functions. Like with traditional real-time search algorithms, such as RTA* [Korf, 1990], the solution obtained theoretically departs from optimality. The reason in our case is that the real-time variant generates the first partial solution sub-tree, whose optimality is based on the “forward” heuristic only (the rollback mechanism not being fully exploited when computing a partial solution). However, the notion of optimality has to be considered in the light of our application: the heuristic functions we have described, which represent narrative concepts (e.g., associated with an actor’s personality, etc.). Departing from optimality in this case does not result in a “poor” solution, but rather in just another story variant. Further, working on explicit AND/OR trees makes obviously possible to design accurate heuristics! Apart from the necessity to interleave planning and execution, there have been efficiency considerations behind the use of a real-time version of AO*. The complexity of search, especially the memory requirements, tends to grow quickly with the depth of trees. We are currently using representations that have a depth of six, just to represent a small fraction of a sitcom episode. This value is consistent with the (generic) plans described by Funge et al. [1999], which have an average depth of seven. However, as we’ll move towards generating longer fragments of an episode, the trees are likely to grow larger. This, together with the increasing number of artificial actors, justifies the real-time version.

4. Character Interaction and Story Emergence

Story generation emerges from the interaction of the actor’s plans. While the story genre prescribes the overall relations between the main characters’ plans, there is no active synchronisation or prescribed dynamic interaction between these plans. The plans are not a priori synchronised: their interaction takes place only through the events taking place in the virtual world.

Let us illustrate this by a few examples. As we have seen, the overall sitcom genre prescribes different plans for Ross and Rachel. Ross’ plan is to seduce Rachel. As such, he must acquire information on her, finding some way of talking to her privately, ensure that she is in a positive mood towards him and eventually ask her out. This would look as a rather simplistic model from a real-world perspective, but is very much in line with the narrative properties of the story genre. This is why the various stages have some natural ordering, which is reflected in the HTN plan formalisation by having the various steps subsumed by an AND node.

A first example can illustrate story generation, in a case largely determined by the main character’s actions. This specific plot is produced from Ross’ generic plan, using heuristics that reflect a strong personality. Hence, he is not shy and not afraid of interrupting conversations or approaching other characters. In the first instance, Ross acquires information about Rachel by reading her diary. In the meantime, Rachel is talking to Monica (Figure 2b). In order to talk privately with Rachel, Ross is simply asking Monica to leave (Figure 2c). After which he is able to ask Rachel out (Figure 2d).

Figure 2: Behaviour Generation (Ross' generic plan)

However, the full potential of story generation derives from the interaction of character's behaviours. Interaction between primary characters is based on one essential principle: compatibility between the main character action and the other character's state. The latter is influenced by awareness. We can illustrate this through a "jealousy" example, which illustrates the interaction between one main character's plan and the second main character's mood. For instance, if Rachel happens to see Ross talking to Phoebe, unaware that he is actually asking her about herself, she might get jealous and mad at Ross, resulting in a comic *quiproquo*. The specific plot generated by the system (Figure 3) is the following. Ross goes to acquire information about Rachel by talking to Phoebe ("ask her friend"). In the meantime, Rachel is reading a book (see Rachel activities, Figure 3b). As the location of the conversation between Ross and Phoebe is visible from where Rachel reads, seeing them makes her jealous (Figure 3c). Still following his plan, Ross' terminal action "ask her out" will fail, as she is jealous.

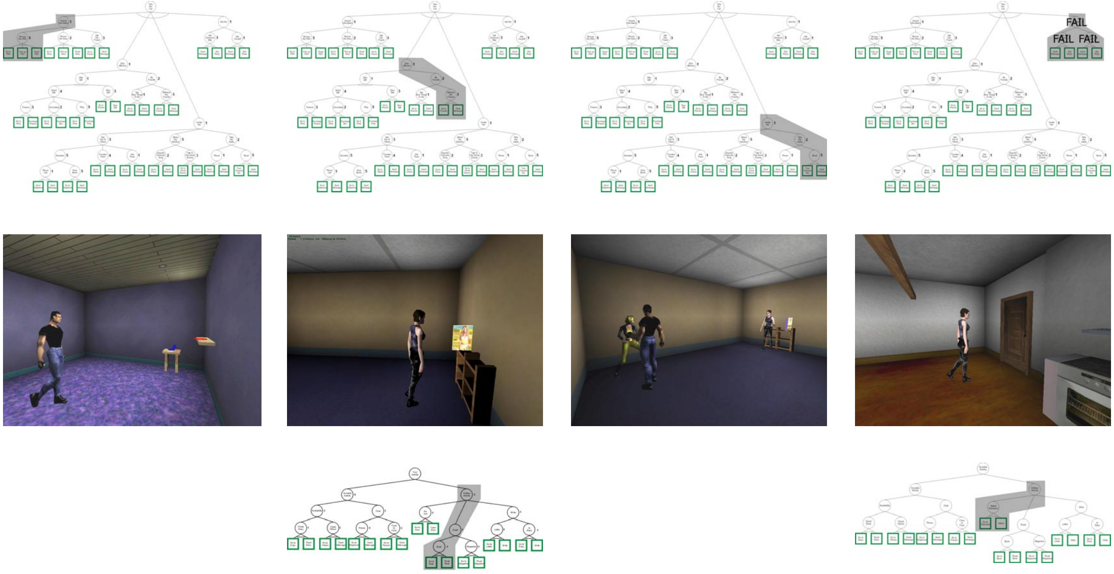


Figure 3: Character Interaction ("Jealousy" plot)

This example also illustrates the specific representation of emotional states, or moods, for the characters. We have defined certain agent states, mostly related to mood value, which condition the character's response to other agent's action. This constitutes an essential element for the story to be understandable, provided the agents moods or emotional states can be perceived by the user. Another aspect is to dramatise the interaction between characters themselves, especially with relation to their emotional states. The kind of animation and camera control used within a game engine would not make easy to express complex non-verbal behaviour manifesting emotions, such as facial expressions or body postures.

We have thus chosen simpler, cartoony modes of expressing feelings, such as blushing or adding expressive icons. As moods can be seen as an alteration of personality, and personality is represented through heuristic functions used in the forward expansion of the (OR) nodes in the AND/OR trees, one simple way of propagating change in mood values is to dynamically alter the heuristic values attached to nodes (this will of course only affect "future" nodes, i.e. nodes yet to be expanded, in accordance with the implicit time ordering). Dynamic alteration of mood values impact on the heuristic evaluation for the nodes yet to be explored in the AND/OR tree. This is illustrated on Figure 4: when Rachel changes mood from "Happy" to "Jealous", the heuristic values attached to nodes in her plan graph are updated accordingly. The new values will favour goals and activities in agreement with her emotional state: for instance she would rather stay alone and read if she is not "Sociable".

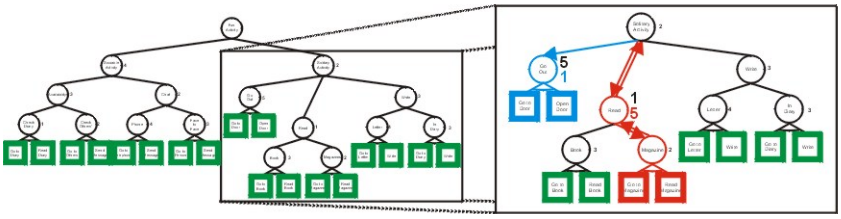


Figure 4: Rachel mood changing to jealous

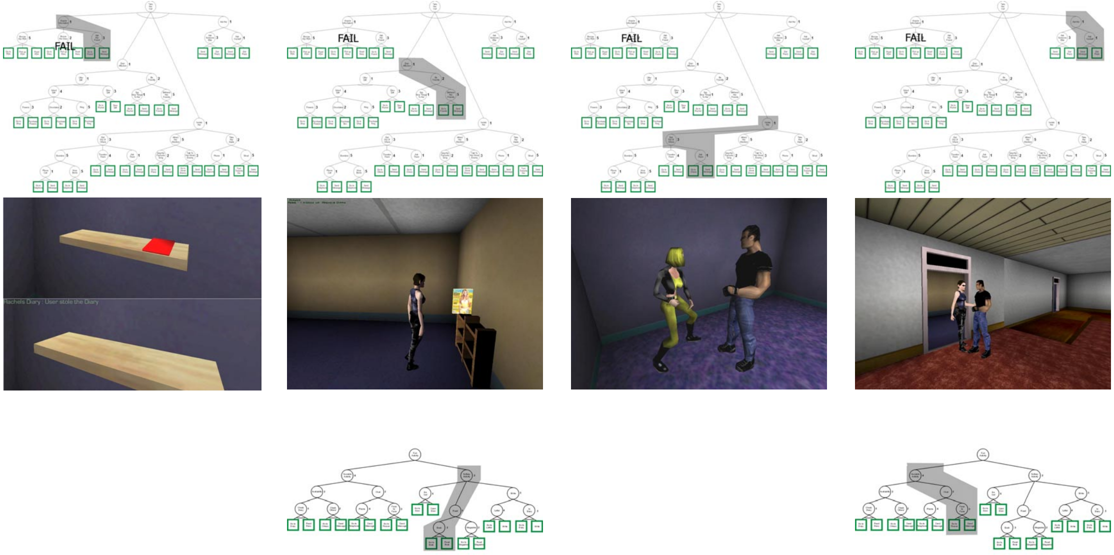
5. User Intervention and Interactive Storytelling

The user observes the story unfolding as it is played by the various characters. At this stage, the story generated by the system is much like a silent movie, though we aim at developing a soundtrack facility later on. In the meantime, we use textual messages to "subtitle" essential actions and situations. The key to user understanding is the dramatisation of actions, the graphic presentation that make the action to convey a narrative meaning. This is also due to the fact that, as agents are directed by story plans, their actions have a narrative meaning: characters are not, for instance, randomly walking on the set, they are always pursuing some active sub-goal, which is often a well-defined stage of the story (and as such identifiable to the user). This is why seeing Ross going for a particular item, such as Rachel's diary, has immediate narrative significance. The user observes the scene by default, where the system's camera focuses automatically on actions carried out by the main characters (Ross, Rachel). However, the user can also explore the stage in an active fashion, visualising the action from a subjective mode in which he controls the camera.

According to the principles we have stated in the introduction, the user i) is allowed anytime interaction and ii) is rather interfering with the action than taking a full part in the story itself as a member of the cast. As such, his involvement is highly focussed and will aim at helping or contrasting specific agents' plans, according to her understanding of the story.

The first one consists in acting on narrative objects, i.e. those objects that are required by the agents as instruments for their actions, such as a diary or a telephone (to acquire information). For instance, the user can steal or hide Rachel's diary, preventing Ross from reading it (see below) or intercept Ross's gift and redirect it to Phoebe, with unpredictable consequences.

This is implemented by resorting to the standard interaction mechanisms in Unreal™, which support interaction with physical objects. Acting in a subjective mode (the actor is embodied through an avatar, though this does not appear as part of the story in first-person mode), the user has access to the same interaction mechanism that the agents have. Many objects on-stage that have narrative relevance are reactive objects: they can be collected or used by all members of cast. Whenever they are collected first by the user, they are unavailable for the actors. It should be noted that in the current implementation, the actors only "know" the default location of any given relevant object and are not able to search their environment for a specific object.



Rachels Diary : User stole the Diary

Figure 5: User interference

As in our current prototype, user intervention takes place through interaction with the set objects, his interventions often interfere with the executability conditions [Geib and Webber, 1993] of terminal actions. Figure 5 illustrates how the user can interfere with the character plan by stealing an object on the set. If, according to his initial plan, the character is going to acquire information on Rachel by reading the diary, the user can contrast that plan by stealing the diary (Figure 5a). This impairs the execution of the 'Read diary' action, after the character has moved to the normal diary location. The fact that the diary is missing is also dramatised, as evidenced on Figure 6.

Ross : I can't find Rachels Diary
Ross : I need to replan a solution

Figure 6: Ross can't find the diary

As the action fails, the search process is resumed to produce an alternative solution for the ‘acquire info’ node (Figure 7), which is to ask one of Rachel’s friends for such information. The Ross character will thus walk to another area of the set to meet “Phoebe” (Figure 8).

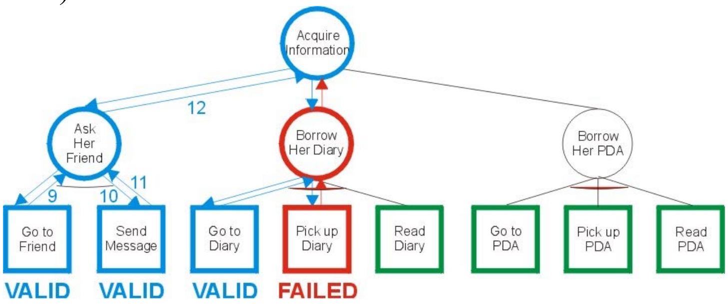


Figure 7: Re-planning following user intervention

Another mode of user intervention, currently under development, consists for the user to provide information or advice directly to the virtual actors using speech recognition [Charles at al., 2001]. This could for instance satisfy one character’s goal, substituting for an information-seeking action (such as “reading Rachel’s diary), or it could try to influence characters by changing their mood state.

Figure 8: Ross talking to Phoebe (alternative plan)

6. Conclusion

We have described a first prototype for interactive storytelling within a character-based approach. The implementation of artificial actors in interactive storytelling is faced with complex technical problems, such as interleaving planning and action, supporting user interaction and representing storytelling concepts.

In this context, we claim that search-based planning provides a practical short-term solution in interactive storytelling and computer games. Further, the use of AO* opens additional perspectives in terms of interaction and counter-planning, as its use has also been described for adversarial search in two-player games (where it shares some properties of SSS* [Stockman, 1979]). This suggests that more complex interactions, involving a larger number of actors, could be explored. Further work on the system will be dedicated to the automatic recognition of emergent episodes in terms of narrative functions. This seems a pre-requisite to gain a better understanding into narrative representation and narrative emergence by experimenting with the system.

Acknowledgements

Eric Jacopin is thanked for his advice on AI planning formalisms: any remaining misconceptions are the authors’ sole responsibility.

References

Barthes, R. 1966. Introduction a l'Analyse Structurale des Récits (*in French*), *Communications*, 8, pp. 1-27.

Bonet, B. and Geffner, H, 1999. Planning as Heuristic Search: New Results. *Proceedings of ECP'99*, pp. 360-372.

Charles, F., Mead, S. and Cavazza, M., 2001. User Intervention in Virtual Interactive Storytelling. *Proceedings of VRIC 2001*, Laval, France, to appear.

Dautenhahn, K., 1998. Story-Telling in Virtual Environments, *ECAI'98 Workshop on Intelligent Virtual Environments*, Brighton, UK, 1998.

Funge, J., Tu, X., and Terzopoulos, D., 1999. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. *Proceedings of SIGGRAPH'99*, Los Angeles (USA), pp. 29-38.

Geib, C. and Webber, B., 1993 A consequence of incorporating intentions in means-end planning. *Working Notes – AAAI Spring Symposium Series: Foundations of Automatic Planning: The Classical Approach and Beyond*. AAAI Press.

Geib, C., 1994. The intentional planning system: Itplans. *Proceedings of the 2nd Artificial Intelligence Planning Systems Conference*, pp. 55-64.

Knight, K. and Rich, E., 1991. *Artificial Intelligence*, 2nd Edition. McGraw Hill.

Korf, R.E., 1990. Real-time heuristic search. *Artificial Intelligence*, 42:2-3, pp. 189-211.

Kurlander, D. and Ling, D.T., 1995. Planning-Based Control of Interface Animation. *Proceedings of the CHI'95 Conference*, Denver, ACM Press.

Mateas, M., 1997. *An Oz-Centric Review of Interactive Drama and Believable Agents*. Technical Report CMU-CS-97-156, Department of Computer Science, Carnegie Mellon University, Pittsburgh, USA.

Nakatsu, R. and Tosa, N., 1999. Interactive Movies, In: B. Furht (Ed*), Handbook of Internet and Multimedia – Systems and applications*, CRC Press and IEEE Press.

Nilsson, N.J., 1980. *Principles of Artificial Intelligence*. Palo Alto, CA. Tioga Publishing Company.

Nilsson, N.J., 1998. *Artificial Intelligence: A New Synthesis*. San Francisco, Morgan Kaufmann.

Pearl, J., 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading (Massachusetts), Addison-Wesley, 1984.

Pemberton, J.C. and Korf, R.E., 1994. Incremental Search Algorithms for Real-Time Decision Making. *Proceedings of the 2nd Artificial Intelligence Planning Systems Conference* (AIPS-94).

Perlin, K.and Goldberg, A., 1995. Improv: A System for Scripting Interactive Actors in Virtual Worlds. *Proceedings of SIGGRAPH'95*, New Orleans (USA)..

Propp, V., 1968. *Morphology of the Folktale*. University of Texas Press:Austin and London.

Schank, R.C. and Abelson, R.P., 1977. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Hillsdale (NJ): Lawrence Erlbaum.

Sgouros, N.M., Papakonstantinou, G. and Tsanakas, P., 1996. A Framework for Plot Control in Interactive Story Systems, *Proceedings AAAI'96*, Portland, AAAI Press, 1996.

Stockman, G.C., 1979. A Minimax Algorithm Better than Alpha-Beta? *Artificial Intelligence*, 12, pp. 179-196.

Tsuneto, R., Nau, D. and Hendler, J., 1997. Plan-Refinement Strategies and Search-Space Size. *Proceedings of the European Conference on Planning*, pp. 414-426..

Webber, B.N., Badler, N.I., Di Eugenio, B., Geib, C., Levison, L., and Moore, M., 1994. *Instructions, Intentions and Expectations*, IRCS Technical Report 94-01, University of Pennsylvania.

Young, R.M., 2000. *Creating Interactive Narrative Structures: The Potential for AI Approaches.* AAAI Spring Symposium in Artificial Intelligence and Interactive Entertainment*, AAAI Press, 2000.*