PLANNING CHARACTERS' BEHAVIOUR IN 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 the behaviour of artificial actors in the context of interactive storytelling. We have developed a fully implemented prototype based on the Unreal Tournament™ 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 in 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. In this context, the narrative equivalent of a character's behaviour consists in its role. The set of possible roles for a given actor is represented as a Hierarchical Task Network (HTN). The system uses HTN planning to dynamically generate the character roles, by interleaving planning and execution, which supports dynamic interaction between actors, as well as user intervention in the unfolding plot. 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, HTN Planning, Autonomous Characters, Virtual Humans

1 INTRODUCTION

The development of artificial actors and AI-based animation naturally leads to envision interactive storytelling systems in the near future. A typical interactive storytelling system would be based on autonomous virtual actors that would 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[1], interactive virtual storytelling[2, 3], user-centered plot resolution[4], character-driven storytelling[5, 6]. 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 narrative control and the existence of an explicit narrative representation, the modes of user intervention, the relations between characters and plot, etc.

Our own conception of interactive storytelling is strongly character-centred[6]. As a consequence, it privileges anytime interaction and occasional involvement of the user rather than full immersion. The long-term applications we have in mind 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 results obtained from a proof-of-concept "virtual sitcom" prototype. The prototype we describe has been developed using the Unreal Tournament™ game engine. In addition to its 3D animation capabilities, the Unreal Tournament™ 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. Its use has been previously reported in prototyping interactive storytelling[6]. The system described in this paper has

Based on "AI-based Animation for Interactive Storytelling" by Marc Cavazza, Fred Charles, and Steven J. Mead, which appeared in Proceedings of IEEE Computer Animation 2001

been fully implemented as a set of template C++ classes, and are used as native functions by UnrealScript™, Unreal Tournament™'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 examples of how a specific plot can emerge from the interaction between characters.

2 CHARACTER BEHAVIOUR AND NARRATIVE REPRESENTATION

Plans constitute the most generic description of an artificial actor's behaviour, both in AI-based animation[7, 8, 9] and in interactive storytelling[5, 6]. Most of the plans described are hierarchical plans that relate characters' intentions, goals, or high-level tasks to primitive actions that can be mapped onto animation sequences. There are specific requirements on planning techniques to be used in interactive storytelling. Firstly, they should interleave planning and execution to be able to properly dramatise the elements of the plot and to allow for interaction with the user or other characters. In a similar fashion, because user intervention can cause actions to fail, the system should have re-planning abilities. Finally, the planning formalism should be appropriate for representing narrative knowledge.

The formalisation of narratives can be tracked back to Aristotle's Poetics. In modern times, Propp[10] has founded the formal description of narratives, through the notion of narrative functions. Structuralists like Barthes proposed structured ("stemma-like") representation of narrative episodes, but also most importantly re-introduced agency and the representation of character's actions[11]. Schank and Abelson later proposed scripts and plans as a computational version of the structuralist formalisation of stories[12]. Narrative formalisms are essentially centered around story elements or narrative events. Planning formalisms can be used to represent the whole narrative itself, as described for instance by Young[13]. This kind of representation propagates causality and controls the game of artificial characters as well as user intervention in the story.

However, we have chosen to use plans to represent individual roles for the artificial actors, rather than the global narrative structure. Each actor is associated a plan corresponding to its role or, more precisely, the set of possible role instantiations according to a given storyline. It can be seen as a resource for story generation. Each plan corresponds to the actor's role in a given story instantiation: it represents the plot through a character's behaviour. The plot itself consists in the on-stage integration of the various roles through the situations created by the interactions between actors.

Of course, this choice has implications for the description of narrative content in the planning formalism. While there are no straightforward rules to convert high-level narrative functions into characters' plans, we have attempted to devise 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 final 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, while Rachel's plan just consists in carrying on her daily activities unaware of Ross, this is likely to result in a series of comic situations and *quiproquos*. 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, depending on her mood and sociability.

In our current prototype, four such characters have their behaviours governed by plans. As described above, Ross' plan is entirely directed towards Rachel, who conversely only has a generic plan for her activities, not specifically oriented towards Ross. The other "secondary" characters have independent plans that will govern their interaction with the main characters. This interaction can be expressed in terms of the situations created (e.g., encounters between characters) or in terms of competition for action resources (narrative objects or other characters).

Finally, it is possible to extend the system by adding new characters, provided their roles can be defined. For instance, it is possible to introduce a rival to Ross, whose plan will be essentially similar to his. The role-based approach is thus a modular one, though certain aspects related to situational reasoning will have to be provided. For instance, if two rival characters plan to seduce Rachel, mechanisms should be provided for their mutual awareness either at plan level or at the level of situation awareness.

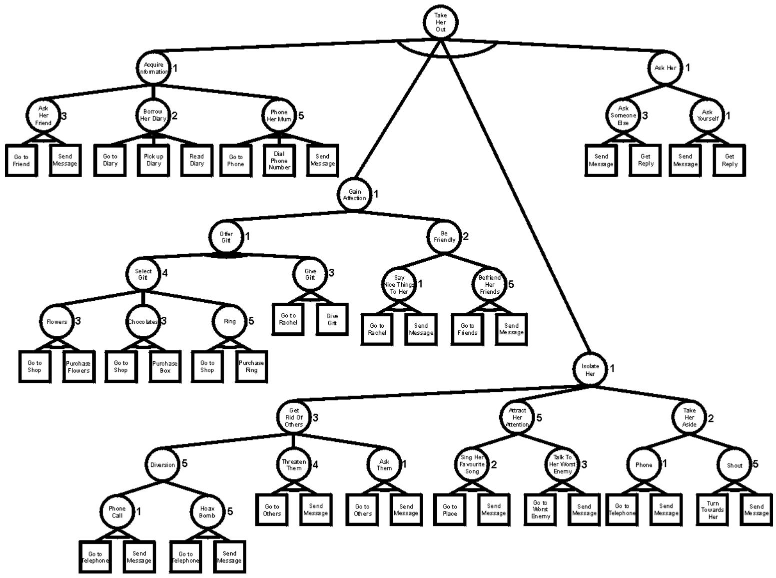

Figure 1: An HTN for the “Ross” Character Role

3 HTN PLANNING FOR BEHAVIOUR GENERATION

As introduced previously, the first step consists in describing the characters’ plan in a planning formalism that will accommodate narrative content. Among various planning formalisms, we have selected Hierarchical Task Networks (HTN), and have implemented a form of HTN planning[14]. In the next sections, we briefly introduce HTN and discuss the rationale for adopting them in the context of characters’ behaviour in interactive storytelling.

HTN can be seen as networks representing (generally ordered) tasks decomposition. The top-level task is also the main goal; each task can be associated a set of methods that decompose it into sub-tasks, which can be further refined until all the actions are primitive actions. HTN are commonly represented as AND/OR graphs (these are generally implicit graphs, though the examples presented in this paper were obtained with explicit graphs).

A single HTN corresponds to several possible decompositions for the main task, because OR nodes correspond to alternative sub-tasks and the sub-tasks subsumed by an AND node can be subject to different orderings. In other words, HTN can be seen as an implicit representation for the set of possible solutions[15]. Our initial descriptions of the various HTN for different characters include a total ordering of the sub-tasks and do not contain variables.

In the present context, each ordered decomposition will constitute the basis for a character’s plan. This can be illustrated on Figure 1: the main goal for Ross is to “ask Rachel out”. This task can be decomposed into various sub-tasks, from “finding information about her” to “offering her gifts”, etc. Each of these sub-tasks is recursively decomposed until primitive actions are obtained. For instance, in the graph 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”. In turn, “Read Rachel’s diary” is decomposed into primitive actions (locating the diary, going to the diary, picking it up, reading it). These are the animation sequences carried out on-stage by the virtual actors that constitute the story in the eye of the spectator. These animations rely on Unreal Tournament™’s built-in mechanisms and animation libraries for the characters.

We originally considered using HTN planning for interactive storytelling as it is generally considered appropriate for knowledge-rich domains, which can provide domain-specific knowledge to assist the planning process[16]. It also appeared that characters’ roles, which serve as a basis for our narrative descriptions, can be naturally represented as tasks decompositions in the HTN planning framework. HTN decomposition is a type of problem reduction that seems well adapted to the description of characters’ behaviour, as it is both expressive and modular. It soon appeared that HTN had many other advantages in this context. In particular, a fundamental difference between STRIPS-style planning and HTN planning is the representation of “desired changes” in the world. Using compound tasks, it is possible to represent a “desired change” that cannot be represented as a single goal or primitive task[17]. This helps to confer a narrative meaning to the “internal” nodes of the HTN. Further, expressivity is enhanced by the fact that ordered task decomposition is both forward and goal-directed at the same time (see below).

In traditional HTN planning, a first decomposition is generated and the corresponding plan is subsequently executed. This is facilitated by the fact that the tasks are generated in the same order as they will later be executed. Several HTN search algorithms have been proposed as part of planners such as UMCP[18] or SHOP[19]. Interactive storytelling requires interleaving planning and execution[13]. We have thus devised a search algorithm to produce a suitable plan from the HTN. Taking advantage from our total ordering assumption and sub-task independence, it searches the HTN depth-first and left-to-right and executes any primitive action it encounters in the process. Backtracking is allowed when primitive actions fail. This search strategy is thus similar to the one described by Smith et al.[20]. In addition, heuristic values are attached to the various sub-tasks, so forward search can make use of these values for selecting a sub-task decomposition.

An essential aspect of HTN planning is that it is based on forward search while being goal-directed at the same time, as the top-level task is the main goal. This contrasts with action-based planning, which is either goal directed (backward search) or based on forward search (from the initial state), as well as with recent search-based planners[21, 22]. An important consequence is that, since the system is planning forward from the initial state and expands the sub-tasks left-to-right, the current state of the world is always known, which makes it possible to perform reasoning about it.

We have chosen to adopt total ordering of sub-tasks for our initial descriptions of roles. Total-order HTN planning precludes the possibility of interleaving sub-tasks from different primitive tasks, thus eliminating tasks interaction to a large extent[14]. In the case of storytelling, the sub-tasks are largely independent as they represent various stages of the story. Decomposability of the problem space derives from the inherent decomposition of the story into various stages or scenes, a classical representation for stories[12]. However, this is largely an empirical finding that we would like to challenge in further experiments.

During the early stages of our work, considering the explicit nature of the HTN, we used the AO* algorithm[23] to produce solution plans. This was also based on results from Tsuneto et al.[24], who showed that when sub-goals are independent, an HTN can be directly searched for a solution without having to generate a state-space graph (i.e. without resorting to serialisation). We are now shifting to more standard HTN planning algorithms so as to be able to benefit from recent developments in the areas and make use of constraints and variables in the graphs.

In interactive storytelling, several actors, or the user himself, might interfere with one agent's plans, causing its actions to fail. The main mechanism for this kind of interaction is competition for action resources, these resources being pre-conditions for actions. In this case, the story can only carry forward if the character has re-planning capabilities. When planning and execution are interleaved, re-planning can take place through direct backtracking in the HTN using the search strategy described above. Ross is executing the sub-task of finding a gift for Rachel, and has opted for flowers. However, having observed this, the user goes on to steal the bouquet of roses (Figure 2b). Ross' action of getting the roses fails. Failure of a terminal action is back-propagated to its parent task ("getting a bouquet for Rachel"), which is marked as failed. The search process will then backtrack to generate an alternative decomposition until a new primitive action is generated and executed, in this case, "getting Rachel a box of chocolates" (Figure 2c).

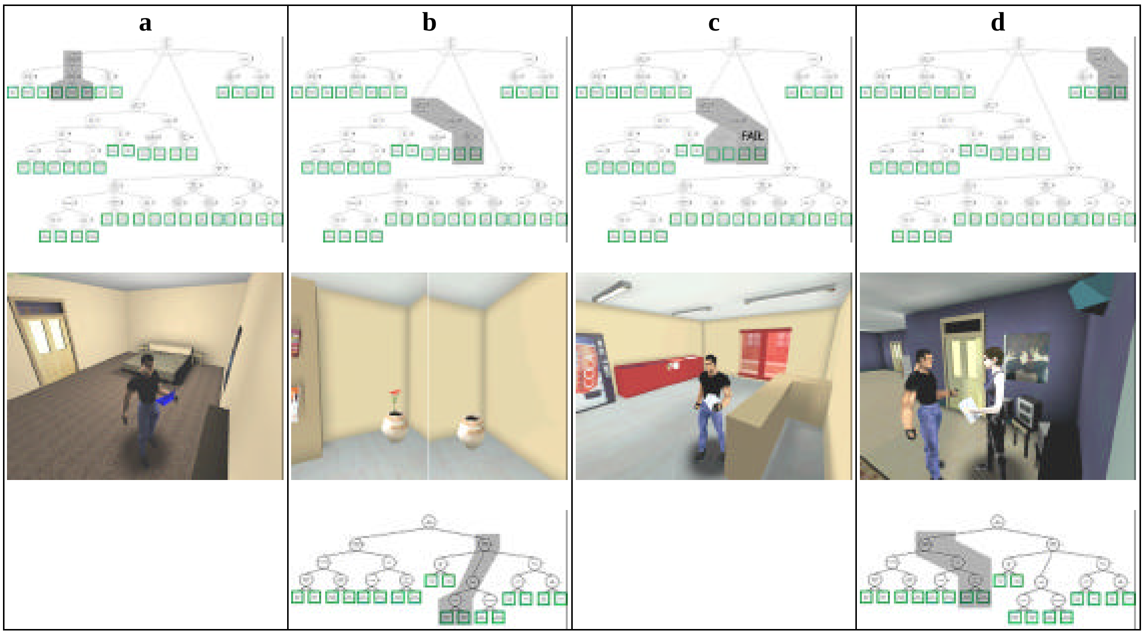

Figure 2: User Interference

In any case, failed primitive actions (looking for an object that is missing, giving an unanswered phone call, etc.) cannot be undone, as they have been played on stage. Action failure is indeed part of the story itself, as well as an important aspect of interactivity. 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.

The complexity of search, especially the memory requirements, tends to grow quickly with the depth of the HTN. We are currently using representations that have a depth of six/seven, just to represent a small fraction of a sitcom episode. This value is consistent with the (generic) plans described by Funge et al.[25], which have an average depth of seven. However, as we'll move towards generating longer fragments of an episode, the system will have to manipulate significantly larger HTN. There are several factors that can contribute to limit the complexity of HTN planning, in particular total ordering and absence of interleaving between sub-tasks. The computational complexity of HTN planning has been studied by Erol et al.[26]. For instance, the total order restriction places HTN planning at the same complexity level as STRIPS-style planning. These theoretical results, together with the empirical performance of HTN planning systems such as SHOP[18] suggest that this approach has a good potential to scale-up.

We have also extended HTN planning with situated reasoning to cope with dynamic situations that arise from agents' interactions. For instance, if Ross encounters Rachel at an early stage of the plot, before he has acquired information on her, his plan has to be interrupted: he can choose to hide from Rachel until he knows enough about her. Another example consists in simple action repair as an alternative to re-planning. For instance, if another actor is competing with action resources, Ross can simply wait for this actor to complete its current action. The combined use of situated reasoning and action repair with search-based planning offers additional flexibility. It is possible to provide a set of narratively relevant repair actions that enhance plot generation without making the definition of roles unnecessarily complex. Situated reasoning was initially introduced by Geib and Webber[8]: here, we make a slightly different use of it, partly due to the fact that the representation behind our HTN planner does not make use on explicit intentions and expectations.

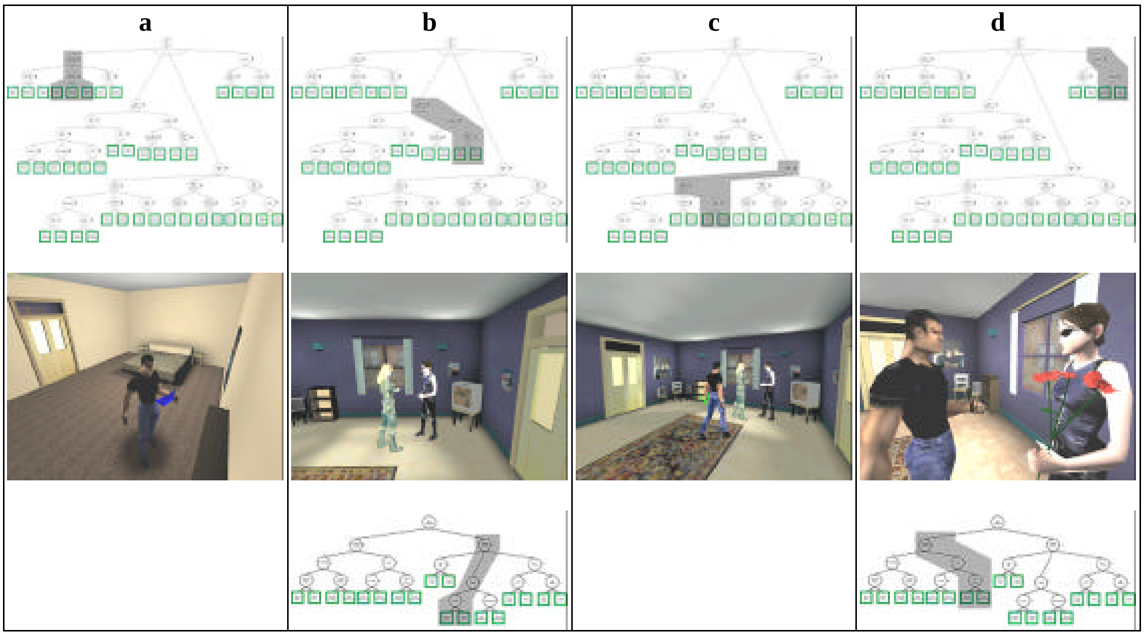

Figure 3: Behaviour Generation

4 CHARACTER INTERACTION AND STORY EMERGENCE

Story generation emerges from the interaction of the actors' plans. While the story genre prescribes the overall relations between the main characters' plans, there is no active synchronisation or pre-defined 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. In other words, there is no explicit representation of narrative functions relating the actions of various characters. These emerge through interaction, though the potential for their occurrence lies in the description of the characters' roles and the mechanisms for situated reasoning.

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 3b). In order to talk privately with Rachel, Ross simply asks Monica to leave (Figure 3c). After which he is able to ask Rachel out (Figure 3d).

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 situation 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 Phoebe about her, she might get jealous and mad at Ross, resulting in a comic *quiproquo*. The specific plot generated by the system (Figure 4) 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's activities, Figure 4b). As the location of the conversation between Ross and Phoebe is visible from where Rachel reads, seeing them makes her jealous (Figure 4c). Still following his plan, Ross' terminal action "ask her out" will fail, as she is jealous.

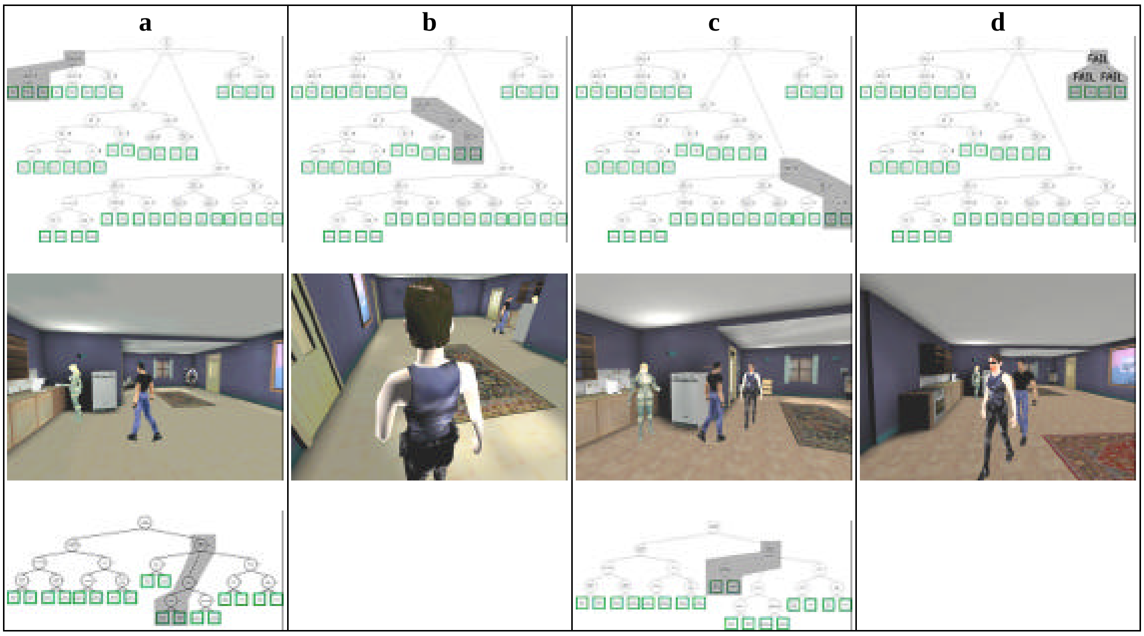

Figure 4: Character Interaction

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 that the agents' moods or emotional states are perceivable 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, cartoon-like 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, 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).

The use of heuristic values to bias search through an action space and obtain narrative effects was originally demonstrated by Weyrauch[27]. Dynamic alteration of mood values impact on the heuristic evaluation for the nodes yet to be explored in the HTN. 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”. The emotional state of the virtual actors can also be altered by the outcome of their actions: for instance action failure would not only trigger re-planning but would also alter the mood of the character.

Though the mechanisms behind story generation are formally deterministic, there are many factors that contribute to story variability. This can be simply illustrated by considering the impact of initial conditions. The various actors are initially spawned at random location on the set. Besides, each character governed by generic activity plans (i.e., in our scenario, all characters but Ross) has his first task selected randomly. The combination of initial positions and initial destinations (determined by the initial task, such as shopping, meeting friends, watching TV, etc.) has a great potential for creating many kind of encounters between characters, or having characters missing one another. For instance, Phoebe’s initial occupation might be to go shopping. Depending on her initial position, she might leave the flat before Ross can ask her information, in which case he would have to devise some other way of learning more about Rachel (see Figure 5). In a similar fashion, depending on Rachel’s initial location and occupation, she might spot Ross talking to Phoebe and mistakenly believe that they are having an affair, leaving the flat in anger.

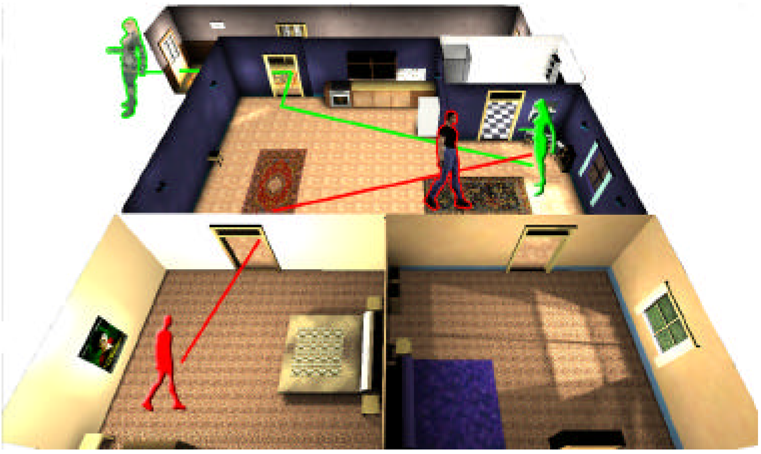

Figure 5: Determinants of Story Variability

5 USER INTERVENTION AND INTERACTIVE STORYTELLING

The user observes the story unfolding as it is being played by the various characters. Currently, the story generated by the system is much like a silent movie, though we aim at developing a soundtrack facility at a later stage. In the meantime, we use textual messages to “subtitle” the most relevant actions and situations. These subtitles include thoughts and actions from a character (e.g. “I can’t find the diary”) and dialogue between characters (e.g. “Would Rachel prefer roses or chocolates?”). The key condition for the user to understand the unfolding story is the dramatisation of actions, the graphic presentation that makes the action convey a narrative meaning. This is due to the fact that, as agents are directed by their role plans, their actions have a narrative purpose: 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 from a default perspective, 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 his understanding of the story. The interaction cycle outlined above is illustrated on Figure 7.

The first mode of intervention 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.

a **b** **c**

Figure 6: Ross can't find the diary

As in our current prototype, user intervention takes place through interaction with the set objects, his interventions often interfere with the executability conditions[8] of terminal actions. Figure 6 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 6b). 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 6c. 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" (Figures 8a, 8b, 8c).

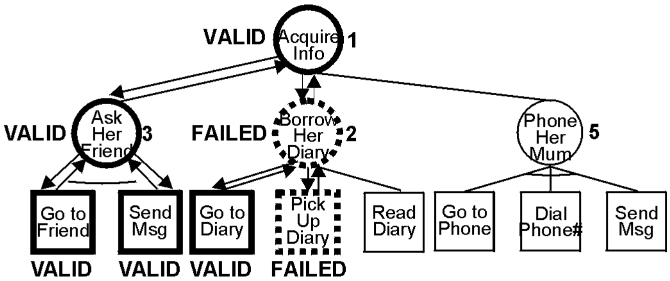

Figure 7: Re-planning following user intervention

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[28, 29]. 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.

a **b** **c**

Figure 8: Ross talking to Phoebe (alternative plan)

6 CONCLUSION

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

In this context, we claim that HTN planning provides a practical solution for interactive storytelling, offering a unified perspective on role definition and dynamic behaviour. The actual number of plots generated is much greater than the size of individual role graphs would suggest - this is because of the impact of backtracking and re-planning, situated reasoning and of the composition of roles. As a consequence, even with a limited number of narrative actions it is possible to obtain a significant variation of plots within the story genre. The independent definition of behaviours corresponding to each actor's role makes also possible to easily include additional actors, and serves as a basis for system scalability. Our initial results were obtained with a traditional plot featuring two principal characters. We have easily extended this first prototype to include four characters each one governed by their own plan. The overhead on including more "active" characters consists in the necessity for reacting to situations that arise from their interactions. Current development include the use of natural language input to influence virtual actors and a re-implementation of HTN planning supporting more complex plans including constraints and variables.

Further work on the system will also be dedicated to the automatic recognition of emergent episodes in terms of narrative functions. This seems a pre-requisite to gain a better understanding of narrative emergence by experimenting with the system.

Acknowledgements

We are indebted to Eric Jacopin for his advice on AI planning formalisms: any remaining misconceptions are the authors' sole responsibility.

The following credits are given for the models used for the actors:
"Ross" created by Brian Collins, converted from Half-Life to Unreal by Kempel.
"Rachel" created by Austin, converted from Half-Life to Unreal by Usaar.
"Phoebe" and "Monica" created by Roger Bacon.

References

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

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

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

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

[5] Mateas M. *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, 1997.

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

[7] Aylett R. Narrative in Virtual Environments: Towards Emergent Narrative. *Papers from the AAAI Fall Symposium on Narrative Intelligence*, Technical Report FS-99-01, AAAI Press, 1999.

[8] Geib C. and Webber B. 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, 1993.

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

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

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

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

[13] Young R. M. Notes on the Use of Plan Structures in the Creation of Interactive Plot in *The Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, Cape Cod, MA, 1999

[14] Nau D.S., Smith S.J.J., and Erol K. Control Strategies in HTN Planning: Theory versus Practice. In *AAAI-98/IAAI-98 Proceedings*, pp. 1127-1133, 1998.

[15] Erol K., Hendler J., Nau D.S., and Tsuneto R. A Critical Look at Critics in HTN Planning. In *IJCAI-95*, 1995.

[16] Kambhampati S. and Hendler J.A. A Validation Structure Based Theory of Plan Modification and Reuse, *Artificial Intelligence*, 55 (2-3), pp. 193-258, June 1992
[17] Erol K., Hendler J., and Nau D.S. HTN Planning: Complexity and Expressivity. In *AAAI-94*, 1994.
[18] Erol K., Hendler J., and Nau D.S. UMCP: A Sound and Complete Procedure for Hierarchical Task-Network Planning. In *Proc. Second International Conf. on AI Planning Systems (AIPS-94)*, pp. 249-254, June, 1994.
[19] Nau D., Cao Y., Lotem A., and Muñoz-Avila H. SHOP: Simple Hierarchical Ordered Planner. In *IJCAI-99*, pp. 968-973, 1999.
[20] Smith S.J.J., Nau D.S., and Throop T. Computer Bridge: A Big Win for AI Planning. *AI Magazine*, 19(2):93-105, 1998.
[21] Bonet B. and Geffner H Planning as Heuristic Search: New Results. *Proceedings of ECP'99*, pp. 360-372, 1999.
[22] Pemberton J.C. and Korf R.E. Incremental Search Algorithms for Real-Time Decision Making. *Proceedings of the 2nd Artificial Intelligence Planning Systems Conference (AIPS-94)*, 1994.
[23] Pearl J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Reading (Massachusetts), Addison-Wesley, 1984.
[24] Tsuneto R., Nau D. and Hendler J. Plan-Refinement Strategies and Search-Space Size. *Proceedings of the European Conference on Planning*, pp. 414-426, 1997.
[25] Funge J., Tu X., and Terzopoulos D. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. *Proceedings of SIGGRAPH'99*, Los Angeles (USA), pp. 29-38, 1999.
[26] Erol K., Hendler J. and Nau D. Complexity Results for Hierarchical Task-Network Planning. *Annals of Mathematics and Artificial Intelligence*, 18:69-93, 1996.
[27] Weyhrauch P. *Guiding Interactive Drama*. PhD Thesis, Carnegie Mellon University, CMU Technical Report CS-97-109, 1997.
[28] Cavazza M., Charles F., and Mead S. J. Non-Instructional Linguistic Communication with Virtual Actors. *Proceedings IEEE International Workshop on Robot and Human Interactive Communication*. Vélizy, France, 2001.
[29] Charles F., Mead S. J. and Cavazza M. User Intervention in Virtual Interactive Storytelling. *Proceedings of VRIC 2001*, Laval, France, 2001.