Agents' Interaction in Virtual 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 fully implemented prototype for interactive storytelling using the Unreal™ engine. Using a sit-com like scenario as an example of how the dynamic interactions between agents and/or the user dramatise the emerging story. Hierarchical Task Networks (HTNs) are formalised using AND/OR graphs, which are used to describe the many possible variations of the story at a sub-goal level, and the set of all behaviours (from a narrative perspective) of the primary actors at a terminal action level. We introduce real-time variant of the heuristic search algorithm AO* that has been implemented to provide a mechanism for planning (and re-planning) and discuss how the chosen heuristic evaluation function is used to describe narrative concepts. We provide early results of several examples of how the same basic plot can have many differing story instantiations as a result of the dynamic interaction within the virtual set and the personalities of the primary characters, and detail the steps required in the plan generation.

1. Introduction: Character-based Storytelling

Interactive Storytelling is one of the most challenging applications for Intelligent Virtual Agents (IVA). Different approaches to Interactive Storytelling have been proposed, which differ by the respective emphasis on various aspects of Interactive Storytelling, such as the user [4][13][18][22], the plot [20] or artificial characters [12] [25]. Character-centred approaches are the most demanding in terms of IVA performance.

There is a strong relation between character and plot in Interactive Storytelling [25]. Characters, as an aspect of narrative [1], are deeply intertwined with plot. If the character can select between various actions at a given stage, the character's choice for action actually dictates the instantiation of the plot. In this context, the plot can be generated dynamically from plans that generate behaviours for each of the characters, depending on the specific circumstances that will result from user intervention [5]. Under this assumption, the global storyline can be implicitly "compiled" in the generic plans describing all possible behaviours for the set of artificial actors.

In this paper, we report early results from a fully implemented Interactive Storytelling prototype. Our system is a character-based system, in which the story emerges from the roles played by the virtual actors, and in particular the dynamic interactions between these roles.

We describe the AI techniques used to implement the behaviour of artificial actors, which are derived from search-based planning. The emphasis of our work in on the relations between narrative descriptions and the dynamic generation of virtual actors' behaviours [4]. We show that dynamic interaction between artificial actors can be a powerful drive for story generation, even in the absence of a centralised plot representation.

2. Planning Techniques for Agent behaviour

Planning is the most generic description of an embodied artificial actor's behaviour, whether from a narrative perspective [12][25] or from a generic "cognitive" perspective [24]. As we are mainly concerned here with the narrative aspects, we should describe how planning techniques can support artificial actors' behaviours in relation with storytelling. This comprises technical constraints, such as response time and the ability for re-planning as well as knowledge representation constraints, i.e. how behaviour can be related to the overall story genre and narrative concepts such as personalities and roles.

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 (HTN), which is formalised as an AND/OR graph. As any task graph, it comprises of plans, goals and actions. These actually correspond to different narrative concepts: plans such as "gaining Rachel's affection", goals such as "isolate Rachel" and actions such as "ask Rachel out".

As we are representing narrative content a priori, our representations are actually explicit graphs. 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 [14]. The process by which a state-space graph is normally produced from a Hierarchical Task Network (HTN) is called *serialisation* [23]. 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 [23]. This makes possible to use search directly on the task network to produce a solution.

Further, there has been recently a renewed interest in search-based planning techniques, as these have demonstrated significant performance on various planning tasks [2][10][17][23]. As the task network for the characters is an AND/OR graph, we naturally use the AO* algorithm [9][15][16] to produce a solution. 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 correspond to a sequence of actions that constitute a specific instantiation of the storyline. These terminal actions give rise to actions in the graphic environment with the corresponding (still interactive) animations taking place.

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 [19].

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 promising sub-graph. For instance, in the tree of Figure 1 corresponding to Ross' plan, 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 (Figure 2).

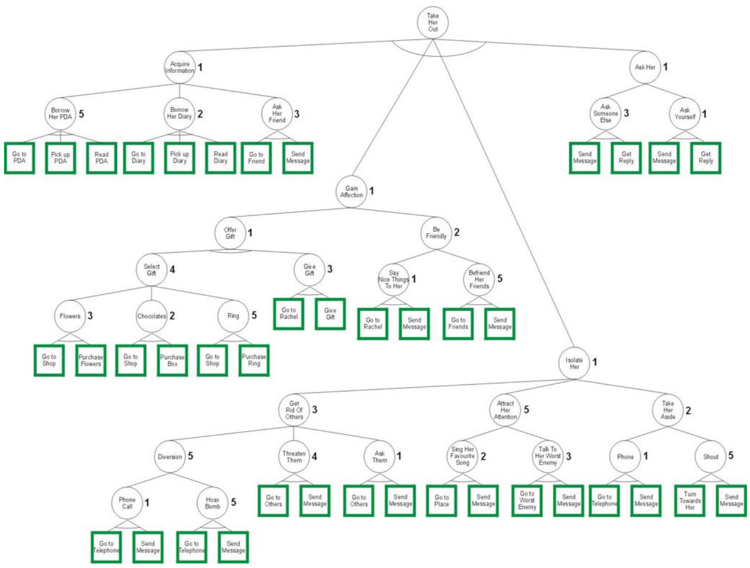

Fig. 1. Plan for the character "Ross"

---

[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.

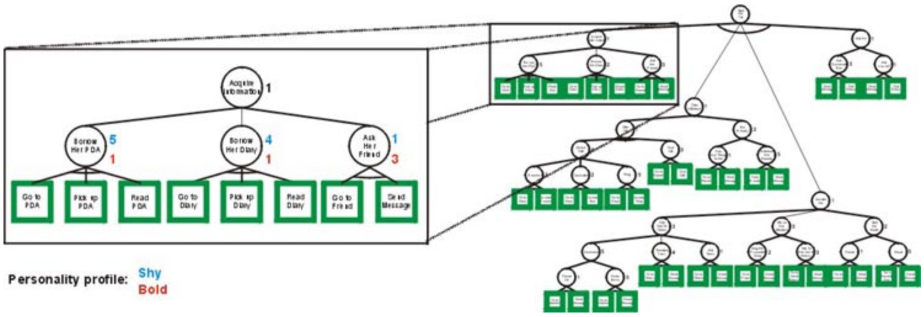

Fig. 2. Ross' personality profiles

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 [16], 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).

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, i.e. the ability to produce a new plan and resume its course of action. 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, as illustrated in Figure 8. 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.

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 [17] using the heuristic part of evaluation functions. Like with traditional real-time search algorithms, such as RTA* [10], 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. [7], which have an average depth of seven. However, future versions of the system will represent more complex episodes and will include a larger number of agents: this justifies the development of a real-time version.

Only principal characters are controlled by plans. This is also consistent with the use of narrative formalisms, which describe events from the perspective a few principal characters. These narrative descriptions were the starting point for defining the contents of our characters' plans [4]. The cast also includes secondary characters that participate to the action. These are "reactive" characters that do not have plan-based narrative representations. Their behaviour obeys simple scripts and they essentially react to external events such as the actions by intelligent characters. They however share mood states with the main characters and these mood states can contribute to create an emotional atmosphere, which can under certain circumstances be propagated to other agents or "sensed" by the principal characters.

Fig. 3. Ross' "active" plan and Rachel's generic pattern of behaviours

3. From Narrative Concepts to Agents' Behaviour

Our whole representational strategy is to represent characters' behaviour from a narrative rather than a cognitive perspective. Another way to express this fact is to say that the artificial actors are acting rather than improvising. The actions they are taking correspond to possible behaviours according to an overall story line, not to generic beliefs, desires or intention[2]. The generative aspects remain subordinated to this storyline [4].

Let us detail Ross' plan. The top-level goal for this episode is to take Rachel out for dinner. This is subdivided into various phases: getting to know Rachel better (gathering information on her taste and hobbies), trying to raise her mood, finding a way to talk to her in private, and finally asking her out. These phases are largely

[2] "Cognitive" approaches would see the agents improvising, with no guarantee that the action so generated would be narratively meaningful.

sequential, that is to say that the top layers of the task network follow an implicit time ordering. In other words, the AND nodes have an implicit temporal interpretation.

These phases actually comprise many variants: for instance, Ross might be too shy to ask Rachel out and will have some of Rachel's friends asking her on his behalf, etc. All of these variants are represented as branches in OR nodes. These actual actions are themselves represented as sub-plans. The lowest-level actions are called *terminal actions*, which interface with the animation procedures in the Unreal™ environment. For instance, a terminal action such as "GoToDiary" will correspond to the Unreal™ script implementing the low-level primitives such as target identification, path planning, etc.

An essential mechanism of story generation is thus the contrast between Ross' and Rachel's plans (Figure 3). Ross' plan is entirely dedicated to seducing Rachel, while Rachel's plan is dedicated to "ordinary" activities. As such, her actions are not strictly speaking goal-oriented, and the plan has to be considered a convenient representation rather than a strict formalisation of her activity. The alternatives encompassed are sometimes unrelated and rather than having a temporal ordering like Ross' plan, correspond to alternative activities. This basic difference in their plan orientation actually accounts for narrative knowledge related to the story genre (sitcom). This sets the conditions for the dynamic interaction between the characters to generate *quiproquos* and comic situations.

Another narrative aspect related to character representation consists in representing their personalities and moods, and the impact these have on the actions they take. These mental states control two main behavioural parameters for artificial actors: their choice of action from the story line and their response to direct actions from other agents. However, personalities and moods are subordinated to the overall narrative structure, as reflected by the plans' contents. In other words, psychological causality is subordinated to narrative causality.

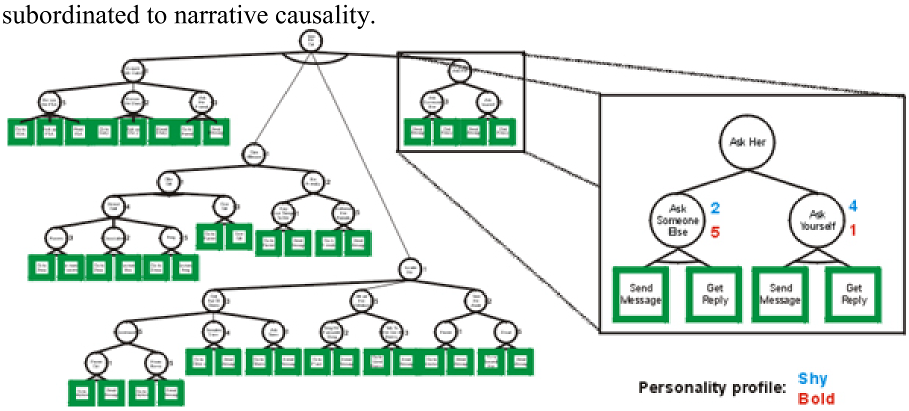

Fig. 4. Ross' personality-driven behaviour

We can illustrate the former point by two examples both from Ross' and Rachel's behaviours. A shy Ross will generally avoid actions that carry potential for conflict or confrontation, such as directly calling Rachel (he will rather ask someone else to ask her out on his behalf, Figure 4). Rachel herself will select her possible activities depending on her mood. Some of these activities are social, some are solitary, and others have her moving away from the set. As such they all have an impact on the story instantiation. Rachel is the best example of a character responding to actions according to her mood. This is even one of the main mechanisms for drama [6] though it essentially plays an instrumental role in our approach.

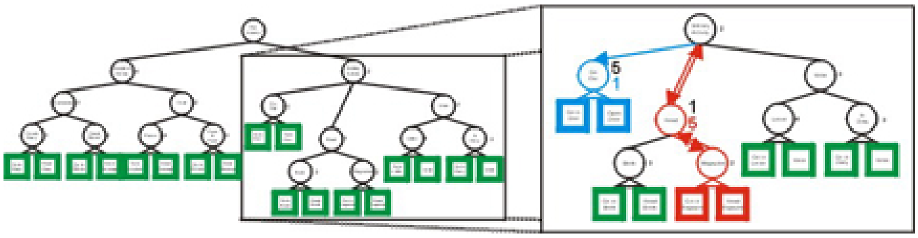

Fig. 5. Rachel's mood changing to jealous

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 graphs. A 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 graph. This is illustrated on Figure 5: 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".

Fig. 6. Dramatisation of Rachel's emotional status

As part of the story believability, it is necessary to make the agents' emotional status visible to the user, so he can understand their interactions. The kind of animation engine we use makes it difficult to represent facial expressions, or detailed non-verbal behaviour, which would also complexify real-time camera control. This is why we had to investigate other ways to dramatise this kind of information. Figure 6 shows one possible solution, which consists in having an "emotion T-shirt" displaying (only when appropriate) the emotional status of a character.

4. Agent Interaction and Story Generation

The basic mechanisms for story generation from agents' behaviours are deterministic. At the "macroscopic" level of the plot as it unfolds on-stage, an important aspect of believability is that the course of action should not be too easily predictable by the user. There are a certain number of factors that contribute to the variability of events: i) the initial random configuration of the artificial actors on stage ii) user intervention, which is by nature non-deterministic, and, most important iii) the dynamic interaction of agents' behaviours.

We should discuss these factors briefly, giving a more extensive description of the dynamic interaction between agents in the next section.

The story starts on stage with the various characters scattered around the set at random positions. The reason why this affects the story can be understood considering the course of actions. Actions have duration, and compete for resources. For instance, Rachel's initial plan is to carry out her normal activities. This plan totally ignores Ross' intentions; it is neither co-operative nor explicitly a counter-plan. If Rachel meets Phoebe on her way to the store, she will stay in the flat longer, giving time for Ross to join them. If she manages to go out, then the story starts with a "near-miss", with consequences on Ross' emotional status that can lead to later blunders, etc. A detailed example in the final section of the paper will also demonstrate the influence of initial configurations on the possibility or "encounters" between the virtual actors.

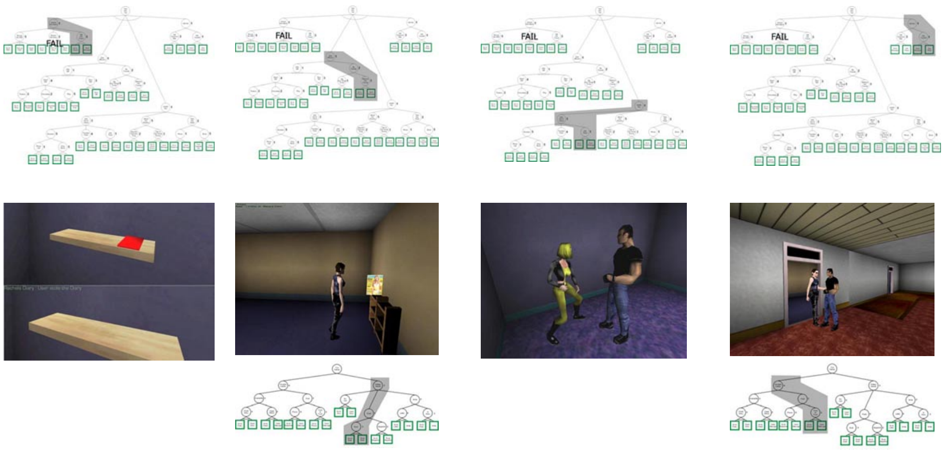

Fig. 7. User interference

The rationale for all user intervention is the desire to alter the plot. Ideally, the goal would be to change the ending of a familiar story line, towards either a happier or more humorous ending. In most cases though, the interference is more character-centred, and consists in contrasting or facilitating the actions of one of the virtual actors to generate comic situations. In our case, Ross is often the main target for user intervention.

User intervention is conditioned by their interpretation of the story. This interpretation is based on the dramatisation of actions. To interfere with the characters' actions, the user must be able to determine what a likely next move could be. In other words, user intervention is strongly dependent on the dramatisation of ongoing events. As the virtual characters are acting rather than improvising, their actions always have narrative meaning. If Ross moves towards an object, this is likely to bear relevance to the story and stealing that object would hence alter the plot.

The main mode of user 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 PDA (to acquire information). For instance, the user can steal or hide Rachel's diary (Figure 7), preventing Ross from reading it (see below) or intercept Ross's gift and redirect it to Phoebe.

Fig. 8. Ross can't find the diary

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.

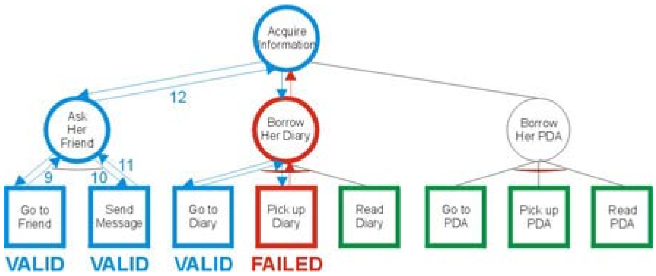

Fig. 9. Re-planning following user intervention

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 7 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 7a). 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 in Figure 8.

As the action fails, the search process is resumed to produce an alternative solution for the 'acquire info' node (Figure 9), 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 10).

Fig. 10. Ross talking to Phoebe (alternative plan)

4.1. Agent-Agent Interactions

Dynamic interaction between characters is an essential element of story generation. There is no synchronisation between the planning/search algorithms of the various actors. All interactions take place through the actions they carry out on stage in the physical environment. As a consequence, plans interleave as a result of their timing and execution.

We should concentrate here on those interactions that affect their respective behaviour in a way that is not explicitly prescribed by the storyboard. By this we mean, interaction between their behaviours through on-stage resources or other actors, rather than direct interaction prescribed as terminal actions of their plans (such as Ross asking Rachel out).

Interaction through resources is a mechanism that is similar to that of human intervention. It conditions the further evolution of the other character's plan, sometimes forcing it to carry out re-planning, like human intervention will do. One characters intervention will affect the other character's behaviour, but not always in the desired direction, i.e. the one that satisfies its own plan.

One example already part of the standard plan is for Ross to be able to talk to Rachel in private. If she is busy, this will imply a certain number of strategies to free her: asking the other actors to leave, calling Rachel or attracting her attention, etc. These situations arise from Rachel activities involving other (secondary) characters. A similar problem derives from some generic knowledge of her activities: Ross can render Rachel available by facilitating some of her activities. For instance, if she is about to go out for shopping, Ross can provide her with the goods she is missing, keeping her in the flat while possibly improving her mood at the same time.

Another illustration of competition for resources represents the dependence on the timing of actions. The initial spatial distribution of actors in the graphic environment implies an influence on actions like Ross wanting to read Rachel's diary, to acquire some relevant information about her. When he reaches the room where the diary is located, Rachel is writing in it. The current terminal action in Ross' sub-plan fails, and a new solution needs to be re-computed to select a new satisfactory sub-goal.

Similar circumstances can occur with the interaction between main characters and secondary characters. When Ross needs to talk to Rachel in private, he may be faced with the situation where she is already chatting with Phoebe. As narrative knowledge is contained in Ross' plan, such as his personality, the story will unfold differently whether Ross is shy, as he would find a way to attract Rachel's attention (e.g. play her favourite song, etc.), or he may be ruthless and would interrupt their conversion with no delay.

Even though elementary scripts govern their behaviour, secondary characters play an important role in story generation, essentially by interacting with the principal characters. Some of the interaction between principal characters is actually mediated by secondary characters. We will provide several examples of this interaction.

An example of this is "mood propagation". If Phoebe and Rachel are busy chatting, Ross, following his plan to isolate Rachel, will interrupt the conversation. He can obviously do so in various ways, either targeting Phoebe or Rachel. One of the options is to chase Phoebe: her emotional mechanisms will be triggered, putting her in a quite upset mood. This mood can be communicated to Rachel, who, upset by Ross' ruthless behaviour will at a further stage not be receptive to his interest in her. The use of a sensory system (Unreal™ engine provides low-level vision and hearing to actors) and a basic rule-based system are combined to provide embodied mechanisms of situation awareness to actors.

Another form of interference is common to the secondary actors and the user: it consists again in competition for action resources. The object that Ross needs, i.e. the telephone can be used by a secondary actor, or more classically, Rachel can be busy talking with another (secondary) character, in which case she is not available for Ross to talk to her. The process by which this competition takes place is twofold: i) their random scripts make them use common resources (e.g. the telephone) ii) the principal character interacts socially with other agents.

5. Results

While the conditions for agent interaction lie in the on-stage spatio-temporal instantiation of the storyline, additional mechanisms are required to recognise these interactions and propagate their consequences. Figure 11 illustrates an entire story instantiation.

> In order to get the information he needs, Ross goes to read Rachel's diary (a). When he approaches the room, he realises that Rachel is actually writing in her diary. Unnoticed by Rachel, he goes to meet Phoebe to ask her about Rachel (b). In the meantime, Rachel has finished writing and decides to have a chat with Phoebe (c). As she arrives to meet Phoebe, she sees her in a joyful conversation with Ross (d). She gets jealous and ostensibly leaves the room (e). Ross sees her leaving and follows her to ask her out, which she refuses (f).

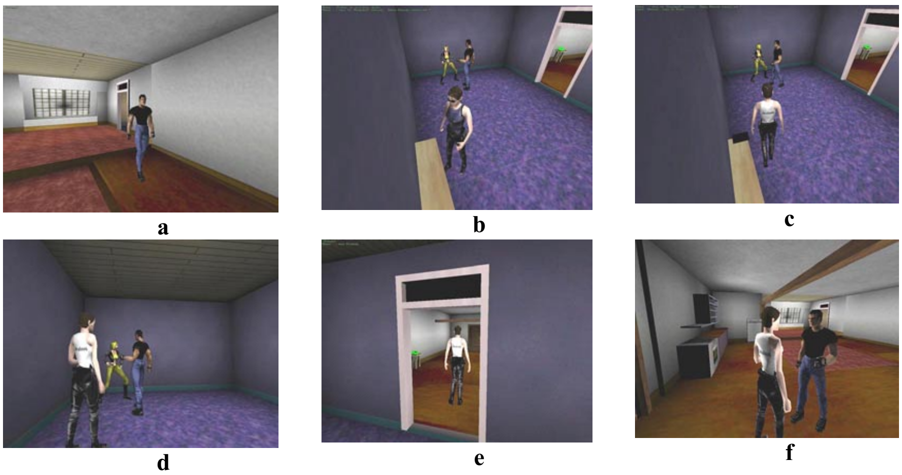

Fig. 11. Storyboard of the "jealousy" scenario

Let us now give a more technical description of these events, by detailing the associated steps in plan generation or terminal actions. Each of the main characters has its own planning system: they are synchronised through Unreal™ low-level mechanisms. Firstly, Ross' plan. The first sub-goal for Ross' plan is to acquire information about Rachel. There are various ways to satisfy this goal in Ross' behaviour representation, and the first one selected is to read her diary. The corresponding script involves going to the diary location and reading it (reading it always succeeds in providing the information). The first part of the script is executed and played on stage. In the meantime, Rachel's plan that governs her spontaneous activity, determines her to write something in her diary. She reaches the diary and starts using it through a durative action (a scripted action which is associated a clock based on the internal Unreal™ clock). When Ross arrives in sight of the diary, the pre-conditions of the action of "reading it" are checked: the diary is in place and that no one else is using it. This pre-condition is not satisfied, hence the second terminal action "ReadDiary" fails, as well as the whole sub-plan. The re-planning produces a new partial solution, which consists in asking Phoebe. Ross then goes to Phoebe's location and starts talking to her. As Phoebe is a reactive actor, she responds directly to Ross' request, in this case positively. In the meantime, Rachel's next occupation is to talk to Phoebe. When she reaches Phoebe, the internal mechanisms will make Rachel aware of the situation where Ross is talking to Phoebe. The pre-conditions for a terminal action involving conversation with another actor is to check whether this actor is free. The jealousy rule is added on top of this check and concerns subjects with which there is a relationship.

The jealousy state is dramatised towards the user by means of an emotive T-shirt (Figure 6), which displays relevant mood states. Internally, the mood state is altered accordingly: all heuristics are revised, and of course, the activity "Chat with Phoebe" fails. Rachel leaves the room. In the same way, Ross' low-level mechanisms will provide situational information that will modify his internal states and influence his sub-plans. Ross will stop talking to Phoebe (terminal action fails) when he realises Rachel is upset, and will then run after her.

To summarise, this example illustrates the interaction of the two main character's plans. Though these plans are designed from global narrative principles (considering the story genre), they are run independently. The particular interactions that take place depend on a number of variable factors, which contribute to the diversity of plots generated. For instance, as we previously discussed, the act that Rachel was able to "surprise" Ross with Phoebe depends on the timing of various actions (her writing in her diary, Ross talking to Phoebe), the time required to walk to Phoebe's location (that depends on the actors initial positions), etc.

6. Conclusion

We have shown that, although actor's behaviours are deterministic, the interaction between actors could considerably contribute towards story variability. This degree of unpredictability conditions the generation of dramatic situations. The character-centred approach has the advantage of being modular and extendable to many actors. Besides, it is not faced with the complex control problems like those of explicit plot representations.

At this stage, the system has been able to generate several relevant variants of the main storyline, mainly based on agents' interaction or user intervention. It is certainly too early to carry out a proper evaluation of the system, which is still under development and is to be tested using more complex storylines. An important question is the relevance or interest of the story variants generated. Relevance, sees as narrative significance, should naturally derive from the original narrative content, which underlie the characters' behaviours: in that sense, relevance is less of a problem than with emergent storytelling approaches. On the other hand, a balance has to be found between the user empowerment (his ability to alter the plot) and the integrity of the original narrative genre from which the storyline was derived. Though, to a large extent, sitcom applications might not be faced with this problem, this aspect is an interesting area for further investigation.

Further work is to be dedicated to developing more complex storylines, scaling up (multiple plans for each actor), narrative function recognition and automated control of camera movements.

Acknowledgements

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

References

1. Barthes, R. 1966. Introduction a l'Analyse Structurale des Récits (in French), Communications, 8, pp. 1-27.
2. Bonet, B. and Geffner, H, 1999. Planning as Heuristic Search: New Results. Proceedings of ECP'99, pp. 360-372.
3. Cavazza, M., Aylett, R., Dautenhahn, K., Fencott, C. and Charles, F., 2000. Interactive Storytelling in Virtual Environments: Building the "Holodeck". Proceedings of VSMM 2000, Gifu, Japan.
4. Cavazza, M., Charles, F., and Mead, S.J., 2001. Characters in Search of an Author: AI-based Virtual Storytelling. First International Conference on Virtual Storytelling, Avignon, France, to appear (2001).
5. Charles, F., Mead, S. and Cavazza, M., 2001. User Intervention in Virtual Interactive Storytelling. Proceedings of VRIC 2001, Laval, France.
6. Dautenhahn, K., 1998. Story-Telling in Virtual Environments, ECAI'98 Workshop on Intelligent Virtual Environments, Brighton, UK.
7. 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.
8. 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.
9. Knight, K. and Rich, E., 1991. Artificial Intelligence, 2nd Edition. McGraw Hill.
10. Korf, R.E., 1990. Real-time heuristic search. Artificial Intelligence, 42:2-3, pp. 189-211.
11. Kurlander, D. and Ling, D.T., 1995. Planning-Based Control of Interface Animation. Proceedings of the CHI'95 Conference, Denver, ACM Press.
12. 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.

13. 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.
14. Nilsson, N.J., 1980. Principles of Artificial Intelligence. Palo Alto, CA. Tioga Publishing Company.
15. Nilsson, N.J., 1998. Artificial Intelligence: A New Synthesis. San Francisco, Morgan Kaufmann.
16. Pearl, J., 1984. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Reading (Massachusetts), Addison-Wesley, 1984.
17. 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).
18. Perlin, K.and Goldberg, A., 1995. Improv: A System for Scripting Interactive Actors in Virtual Worlds. Proceedings of SIGGRAPH'95, New Orleans (USA).
19. Schank, R.C. and Abelson, R.P., 1977. Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures. Hillsdale (NJ): Lawrence Erlbaum.
20. 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.
21. Stockman, G.C., 1979. A Minimax Algorithm Better than Alpha-Beta? Artificial Intelligence, 12, pp. 179-196.
22. Swartout, W.R., Gratch, J., Hill, R., Marsella, S., Rickel, J. and Kyriakakis, C., 2001. Toward the Holodeck: Integrating Graphics, Sound, Character and Story. Proceedings of Autonomous Agents 2001, pp. 409-416.
23. 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.
24. 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.
25. Young, R.M., 2000. Creating Interactive Narrative Structures: The Potential for AI Approaches. AAAI Spring Symposium in Artificial Intelligence and Interactive Entertainment, AAAI Press.