Interacting with Virtual Characters in Interactive Storytelling

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

ABSTRACT

In recent years, several paradigms have emerged for interactive storytelling. In character-based storytelling, plot generation is based on the behaviour of autonomous characters. In this paper, we describe user interaction in a fully-implemented prototype of an interactive storytelling system. We describe the planning techniques used to control autonomous characters, which derive from HTN planning. The hierarchical task network representing a characters' potential behaviour constitute a target for user intervention, both in terms of narrative goals and in terms of physical actions carried out on stage. We introduce two different mechanisms for user interaction: direct physical interaction with virtual objects and interaction with synthetic characters through speech understanding. Physical intervention exists for the user in on-stage interaction through an invisible avatar: this enables him to remove or displace objects of narrative significance that are resources for character's actions, thus causing these actions to fail. Through linguistic intervention, the user can influence the autonomous characters in various ways, by providing them with information that will solve some of their narrative goals, instructing them to take direct action, or giving advice on the most appropriate behaviour. We illustrate these functionalities with examples of system-generated behaviour and conclude with a discussion of scalability issues.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *graph and tree search strategies, heuristic method plan execution, formation, and generation.*

General Terms

Theory, Design, Algorithms, Experimentation, Verification.

Keywords

Interactive Storytelling, Synthetic Characters, Planning, Computer Games, Speech Understanding.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

1. INTRODUCTION

Interactive storytelling is a major endeavour, which has recently attracted substantial research interest, in particular in the area of synthetic actors, as these would play an essential role in the implementation of future interactive storytelling systems. Current research follows a great diversity of approaches, which sometimes overlap, such as: immersive storytelling [1] [2], emergent storytelling [3] [4], interactive authoring of stories [5] [6], plot-based systems [5] [7] [8] and character-based systems [2] [9] [10] [11] [12] [13] [14]. These do not only correspond to different technical solutions to the problem of generating interactive narratives, but also different design paradigms for the user experience itself, in particular in terms of user intervention on the unfolding story.

In this paper, we describe a character-based interactive storytelling approach supporting anytime intervention by the spectator. Our original idea was to implement the "naïve" situation whereby spectators try to influence the story by "shouting" advice at the on-screen characters. In our prototype, the user can intervene in the story from his/her spectator's position, either by uttering advice to the characters, using speech recognition or, because the environment is represented as 3D graphics, interacting physically with the objects on stage.

In the next sections, we give a detailed account of the mechanisms supporting interactivity in our storytelling system. We first describe the planning techniques underlying characters' behaviour and their relation with narrative concepts and representations. We then discuss the main modes of user intervention, direct physical interaction with narrative objects and communication in natural language with the characters. In particular, we relate the modes of intervention to the mechanisms that account for plot variability. We conclude by discussing evaluation and scalability issues for this approach.

2. SYSTEM OVERVIEW

We have implemented several incremental versions of our research prototype. The scenario used is inspired from a popular sitcom, and is based on a romance between the two main characters, with other actors taking part in the story. The rationale for using the sitcom as a narrative genre is that it constitutes a test bed for the generation of alternative endings as well as intermediate situations. This implementation naturally abstracts itself from some characteristic elements of real-world sitcoms (soundtrack, non-verbal attitudes and behaviours) to concentrate on the mechanisms of the dramatic action themselves, such as misunderstandings, quiproquos, failures, etc. Our system takes as its staring point a basic storyline that defines the various characters' roles, which will be dynamically altered by the interaction between characters and user intervention, but will remain within the boundaries of the initial story genre (see Figure 1).

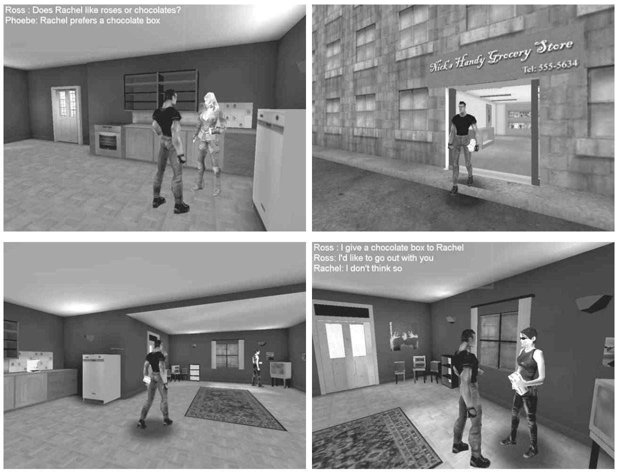

Figure 1. A story instantiation generated by the system: Ross asks Phoebe Rachel's preferences, but Phoebe lies to him.

The graphic environment for our system is based on the Unreal™ computer game engine. Other researchers in the field of interactive storytelling have previously described the use of the same game engine [12] [13], which is increasingly used in non-gaming applications, since the work of [15]. The main advantage of a game engine is to provide both high-quality graphics and a seamless integration of visualisation and interaction with the environment objects. Further, the software architecture offers various modes of integrating software, via C++ plugins or UDP socket interfaces, through which we have integrated a commercial speech recognition system.

3. NARRATIVE REPRESENTATIONS FOR CHARACTER-BASED STORYTELLING

As a general rule, character-based storytelling systems do not represent explicitly narrative knowledge, such as narrative functions or decision points, as in [5] or [7], which could be direct target for user interaction. For instance, in the system described by Sgouros et al. [7], the user is prompted for strategic decision to be made, and narrative causality is maintained via an Assumption-based Truth Maintenance System (ATMS), a process described as "user-centred plot resolution". In the interactive storytelling authoring system of Machado et al. [5], narrative events are generated using a description in terms of narrative functions inspired from Propp [16], which can constitute basic building blocks for the plot.

The "Story Nets" described by Swartout et al. [2] correspond to a plot-like representation of the consequences of user action. However, unlike with user-centered plot resolution [7], these plot models need not be explicit and can be derived from rules operating on key decision points corresponding to user actions [17]. This system integrates aspects from both plot-based and character-based systems. It is however strongly centred on user behaviour and its nominal mode assumes permanent user involvement.

On the other hand, in character-based approaches, the plot is generated by the multiple interactions between autonomous characters. The problem with which character-based systems are generally faced is to ensure that the actions they take are narratively relevant. This corresponds to the narrative control problem and has been studied by Young [13] and Mateas and Stern [18] among others.

In our system, the plot should be mainly driven by the synthetic characters, which is the only approach supporting continuous storytelling with anytime user intervention. In order to reconcile the character-based approach with the problem of narrative control, we describe characters' behaviours in terms of roles, i.e. a narrative representation of their goals and corresponding actions.

For instance, our principal character, Ross, plans to seduce the character Rachel. His role can be described into greater details as a refinement of this high-level goal. Such a refinement will define the various steps he'll take in seducing Rachel, such as acquiring information about her, gaining her friendship, finding ways to talk to her in private, offering her gifts, inviting her out, etc. These also correspond, at its first level of refinement, to the various stages of a (yet linear) story. However, this role representation also includes, as it is refined, a large set of alternative solutions at each further level. The terminal nodes correspond to the final actions actually "played" on-stage through 3D animation of the synthetic characters. They consist in interactions with on-stage objects (watching TV, reading a book, buying gifts, making/drinking coffee…) and other members of the cast (talking, socialising, etc.).

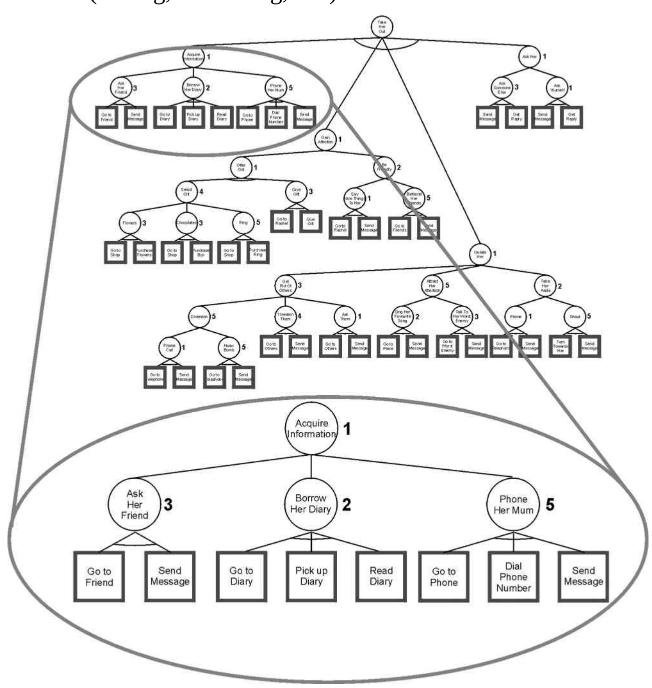

Figure 2. HTN representation for character behaviour.

The characters' roles can thus be represented in a consistent fashion as Hierarchical Task Networks (HTN): this represents an actor's potential contribution to the overall plot (see Figure 2). A single HTN corresponds to several possible decompositions for the main task: in other words, an HTN can be seen as an implicit representation for the set of possible solutions (Erol et al., 1995). This naturally led us to investigating the use of HTN planning techniques to underlie characters' behaviour [12] [13]. In the next section, we describe our approach to planning for characters' narrative behaviours and how these have been extended to incorporate user intervention.

4. PLAN-BASED BEHAVIOURS IN INTERACTIVE STORYTELLING

There is a broad agreement on the use of planning techniques for describing high-level behaviour of autonomous agents embodied in virtual environments, both for task-based simulation [19] [20] and for character-based storytelling [12].

Our description of characters' roles as HTNs naturally led to use these as a starting point for the implementation of a planning system. HTN-based planning, also known as task-decomposition planning, is among the oldest approaches for providing domain-specific knowledge to a planning system. While in the generic case HTN planning may be faced with practical difficulties [21], this approach is considered appropriate for knowledge-rich domains, which can provide applications-specific knowledge to assist plan generation [22]. Interactive Storytelling constitutes such a knowledge-rich application, not least because of the authoring process involved in the description of the baseline story. Besides, there has been a renewed interest in recent years for HTN planning [23], which has demonstrated state-of-the-art performance on a number of benchmarks.

Interactive Storytelling requires interleaving planning with execution. We have devised a search algorithm that produces a suitable plan form the HTN. Taking advantage from our total ordering assumption and sub-task independence, it searches the HTN depth-first left-to-right and executes any primitive action that is generated, or at least attempts to execute it in the virtual stage. Backtracking is allowed when these actions fail (e.g. because of the intervention of other agents or the user). This search strategy is thus essentially similar to the one described by Smith et al. [24]. In addition, heuristic values attached to the various sub-tasks, so forward search can make use of these values for selecting a sub-task decomposition (this is similar to the use of heuristics described by Weyhrauch [25] to "bias" a story instantiation). These heuristic values are used to represent narrative concepts as well. Namely, the various tasks are associated features that index them on some narrative dimension (such as the sociable nature of an activity, or the rudeness of a behaviour), which in turn are converted into heuristic values on these dimensions. Using these heuristics according to his personality and emotional status, a character will give preference to different tasks. These heuristics can be altered dynamically, which in turns modifies subsequent action selection in the character's plan. For instance, Rachel may change mood because some action by Ross has upset her; the consequence is that she would abandon social activities for solitary ones.

Another 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. An important consequence is that, since the system is planning forward from the initial state and expands sub-tasks left to right, the current state of the world is always known, in this case the current stage reached by the plot. We have adopted total ordering of sub-tasks for the initial description of roles. Total-order HTN planning precludes the possibility of interleaving sub-tasks from different tasks, thus eliminating task interaction to a large extent [23]. In the case of storytelling, sub-task independence is an hypothesis derived from the inherent decomposition of a plot into various scenes, though with the additional simplifying assumption that there are no parallel storylines.

There are however additional requirements for planning techniques that control synthetic actors. The environment of the synthetic characters is by nature a dynamic one: the world in which they evolve might constantly change under the influence of other characters or due to user intervention. This would traditionally call for an approach interleaving planning and execution, so that the actions taken are constantly adapted to the current situation. In addition, the action taken by an actor may fail due to external factors, not least user intervention. The latter requires that characters' behaviour incorporate re-planning abilities. As we will see in section 5, these features also support the interactive aspects of storytelling, allowing user intervention to trigger the generation of new behaviours and the corresponding evolution of the plot.

The behaviours for the various characters, corresponding to their individual roles, are defined independently as HTNs. Their integration takes place through the spatial environment in which they all carry out their actions. As a consequence, their on-stage interactions generate a whole range of situations not explicitly described in their original roles.

Examples of such situations obtained with the system are:

1. Ross wants to steal Rachel's diary but she is using it herself, or Phoebe is in the same room, preventing him from stealing it
2. Ross wants to talk to Phoebe about Rachel, but she is busy talking to Monica
3. Ross bumps into Rachel at an early stage of the story, where he has not yet obtained information about her
4. Ross talks to Phoebe but the scene is witnessed by Rachel

Figure 3. Dramatisation of action repair.

These "bottom-up" situations illustrate why the characters' behaviour cannot be solely determined by their top-down planner, in order to be realistic. Situations 1 and 2 would normally lead to re-planning, while more convenient solutions can be devised, such as action repair [26]. In example 1 for instance, Ross could just wait for Rachel to leave, which would restore the executability conditions of the "read_diary" action (see Figure 3). Examples 3 and 4 represent situations that should be actively avoided by the character. A practical solution consists in using situated reasoning, implemented as sub-plans. These are triggered by rules recognising the potential occurrence of such situations and return active post-conditions to the initial plans when it resumes. These mechanisms are further described in [27]. Finally, characters also exhibit reactive behaviour based on some situations: in some cases Rachel can get jealous if she sees Ross in sustained conversation with another female character, or Phoebe can get upset if Ross interrupts her. Reactive behaviours can directly alter the character's plans or trigger scripted response (such as leaving the room). In most cases, though, the output of reactive behaviour is generally to alter the emotional response of the reacting character, which in turns affects its subsequent role. Altering the mood value is equivalent to dynamically changing the heuristic coefficients attached to certain activities. Hence, emotional representations, however simple, play an important role in the story's consistency by relating character behaviour to some personality variables.

Even though the individual mechanisms for actors' behaviour are fairly deterministic, the overall plot generated is not generally predictable by the spectator. Several mechanisms have been incorporated to support, such as the random allocation of characters on-stage, which together with the duration of their actions, greatly affects the probability for encounters, which is a major determinant of plot variability.

The important conclusion is that, while most user interaction takes place through the characters' top-down plans, every mechanism supporting an agent's behaviour is a potential target for user intervention. This will be further discussed in section 6.

5. SYSTEM ARCHITECTURE

The system has been implemented using the Unreal™ game engine as a development environment. The implementation philosophy, like in previous behavioural animation systems is to go from high-level planning to lower-level actions down to animation sequences (which in our case are keyframe animation, but can be interrupted at anytime in case of re-planning).

The game engine offers an API via its scripting language, UnrealScript. Using this scripting language, it is possible to define new actions out of basic primitives provided by the engine; for instance, offering a gift, which consists in passing an object from one character to another. The implementation of an elementary action comprises the updating of graphic data structures (e.g. the object list of a given character or of the environment itself) plus the associated keyframe animation played in the graphic environment.

Characters' roles are generated from HTN plans in the following way. Each character's plan interleaves planning and execution; the lowest-level operators of each plan are carried out in the environment in the form of Unreal™ actions (Figure 4, 1-2), and the action outcome is then passed back to the planner (Figure 4, 3). In terms of architecture the planning component is a C++ module, integrated in the game engine using a dynamically linked library (.dll), which interfaces with the graphic environment via the actions' representation layer programmed in UnrealScript. Similarly, changes taking place in the environment are analysed in this layer and passed back to the planner (Figure 4, 3).

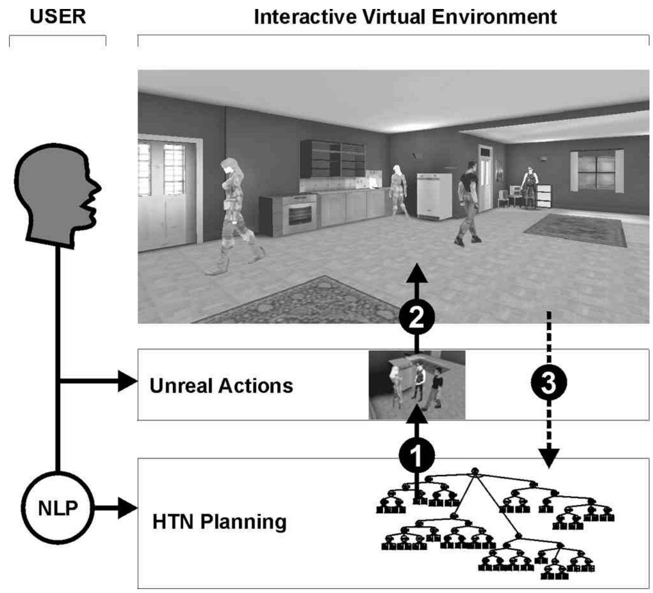

Figure 4. System architecture.

6. PHYSICAL INTERVENTION ON THE VIRTUAL STAGE

The user is a spectator of the unfolding 3D animation corresponding to the generated story, but he can freely explore the stage, being himself embodied through an invisible avatar. This makes it possible for him to interfere directly with the course of action by "physical" intervention on stage. In our current system, physical interaction is limited to narrative objects. The user can remove objects from the stage or change their location, but cannot physically interfere with the actors, for instance by preventing them to enter a room. This is meant to be consistent with the spectator-based approach and its rule of minimal involvement.

Many on-stage objects appear as affordances, i.e. candidates for user interaction. This can be signalled either by their intrinsic narrative significance or by their use by the synthetic characters themselves. The former case is referred to as a "dispatcher" in modern narratology [28]: a dispatcher is an object to which choice is associated, triggering narrative consequences. For instance, in our example scenario, roses and the chocolate box, the potential gifts for Rachel, bear such properties and are a natural target for user interaction. Dispatchers can also be signalled dynamically. As the characters are acting rather than improvising, their actions have direct narrative significance. Hence, if Ross directs himself towards an object, such as Rachel's diary or a telephone, this object acquires narrative relevance and becomes a potential target for user interaction.

Other on-stage objects play a role in the behaviour and most importantly the spatial localisation of the virtual actors. Coffee machines or TV sets are used by the characters: if the user steals the coffee machine that Phoebe was about to use, she would re-plan some other activity, which might take her to another location on the stage. As we have seen, moving to another location can have significant narrative consequences.

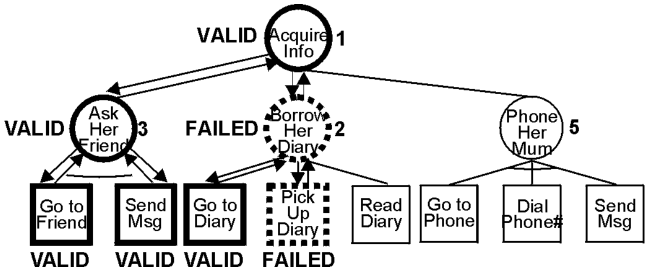

Figure 5. Re-planning on action failure.

From an implementation perspective, actions that are part of the character's plans are associated executability conditions, which include the availability of some resources. For instance, Ross can only read Rachel's diary if it is in the room and Phoebe will only make coffee if the coffee machine is at its usual place. Physical user intervention thus consists in causing character's action to fail by altering their executability conditions. Action failure will in turn trigger re-planning. For instance, Figure 5 shows a fragment of Ross' plan for acquiring information about Rachel. His initial plan consists in reading Rachel's diary, but the user has stolen it. On reaching the diary's default location Ross realises that it is missing and needs to re-plan a solution to find information about Rachel, which in this case consists in asking Phoebe. This is implemented using the search mechanism of our HTN planner by back-propagating the failure of the action "read_diary" to the corresponding sub-goal, so search will backtrack and produce an alternative solution. From a narrative perspective, the user has contrasted Ross' visible goal. But, apart from the immediate amusement of doing so, because failure of Ross' action is dramatised and part of the plot (see Figure 6), the real impact lies in the long-term consequences of the resulting situations. For instance, in the above example, when asking Phoebe about Rachel, Ross might be seen by Rachel, who would misunderstand the situation and become jealous!

Figure 6. Dramatisation of action failure.

This aspect becomes more obvious if considering the interaction with objects used by secondary characters in their normal activities. Phoebe's coffee machine does not have the narrative significance of Rachel's potential gifts; however, displacing it can have serious consequences as well, as she would move on stage and might not be available to answer Ross, or could meet Rachel. While this has proven to be a powerful mechanism for story generation, at this early stage we have not explored its impact in terms of user experience.

7. NATURAL LANGUAGE INTERACTION WITH AUTONOMOUS CHARACTERS

Natural language intervention in interactive storytelling strongly depends on the storytelling paradigm adopted. For instance, permanent user involvement, e.g. in immersive storytelling or training systems [2], requires linguistic interaction to be part of the story itself. This most naturally calls for dialogue-based interaction, as described for instance by Traum and Rickel [29] for the same project.

Our own approach being based on a user-as-spectator paradigm, the user interventions, including speech input are essentially brief and can occur at anytime. They essentially take the form of instructions or advice [19]. Speech input should be tailored to our interactive storytelling context, in which the user influences virtual characters, in order to implement a consistent user experience. For instance, the utterance will often start with the name of the addressee, as in "Ross, be nice to Monica", not only to identify the relevant character but also to establish a simple relation between the user and the character he is influencing. Also, the speech guidance should naturally be in line with the various stages of the plot and correspond to narrative actions and situations. The user can become acquainted with the possibilities of intervention either by being introduced to the overall storyline or, as otherwise suggested by Mateas and Stern [18], through repeated use of the storytelling system.

There has been extensive research in the use of natural language instructions for virtual actors. Webber et al. [19] have laid out the foundations of relating natural language instruction to plan-based high-level behaviour for embodied virtual agents. They have also provided a classification of natural language instructions in terms of their effects. Bindiganavale et al. [30] have described the use of instructions and advice to influence the dynamic behaviour of autonomous agents when dealing with certain situations (checkpoint training). Though these are not specifically addressing storytelling, many of these results can be adapted to a narrative context.

We have incorporated an off-the-shelf system, the EAR™ SDK from Babel Technologies™ into our prototype, which has been integrated with the Unreal™ engine using dynamically linked libraries like for the HTN system. The EAR™ SDK supports speaker-independent input and allows for the definition of flexible recognition grammars that include optional sequences and joker characters. This makes possible to implement various paradigms for speech recognition, from full utterance recognition to multi-keyword spotting. At this stage we are experimenting with a recognition grammar with optional sequences for added flexibility and a small test vocabulary (< 100 words), which includes the main actions and narrative objects, as well as some situations.

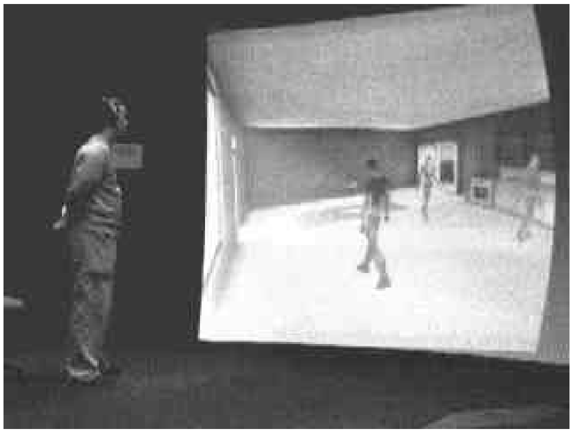

Figure 7. Situational advice: "Ross, don't let Rachel see you with Phoebe".

At this stage the natural language interpretation of user input is based on simple template matching. We have defined templates for several categories of advice, such as: prescribed action ("talk to Phoebe"), provision of information to an actor ("the diary is in the living room", "Rachel prefers chocolates"), generic and specific advice ("don't be rude", "be nice to Phoebe") and situational advice ("don't let Rachel see you with Phoebe" (see Figure 7)), etc.

The instantiation of the template's slots is carried out from simple procedural Finite-State Transition Network parsing of the relevant recognised elements. Consistency checking is based on templates that contain role structures for a certain number of key narrative actions that speech input is supposed to influence. These are based on selectional restrictions for the various slots of a given template. For instance, the advisee is often the main character, especially when doctrine elements are involved.

The selection of the relevant candidate template is determined by the semantic categories of verbs or action markers in the sentence, which are used as heuristics to identify the best template. It can be noted that there is no obvious mapping between the surface form and the interpretation in terms of narrative influence. For instance, "talk to Monica" is interpreted as a direct suggestion for action (which will solve a sub-goal such as obtaining information about Rachel), while "don't talk to Phoebe" is more of a global advice, which should generate situated reasoning whose result is to try to avoid Phoebe. As a generic rule, though, it would appear that most negative statements consist in advice or "doctrine" statements [19].

In our first series of test, we have been essentially focusing on advice related to characters' behaviour, as they have the most dramatic effect, and also as interaction with objects is often the remit of physical intervention on stage.

Overall, we have identified various forms of natural language intervention, such as: the provision of information to an actor (including conspicuously false information), direct instruction for action, warnings, and generic advice on the character's behaviour.

In the next section, we give some examples of linguistic interaction and relate these to the mechanisms by which their effects on characters behaviours and on the plot are actually implemented.

7.1 EXAMPLES

The direct provision of information can solve a character's sub-goal: for instance, if, at an early stage of the plot, Ross is acquiring information about Rachel's preferences, he can be helped by the user, who would suggest that "Rachel prefers chocolates". The provision of such information has multiple effects: besides directly assisting the progression of the plot, it also prevents certain situations that have potentially a narrative impact (such as an encounter between Ross and Phoebe) from emerging. From an implementation perspective, sub-goals in the HTN are labelled according to different categories, such as information_goals. When these goals are active, they are checked against new information input from the NL interface and are marked as solved if the corresponding information matches the sub-goal content.

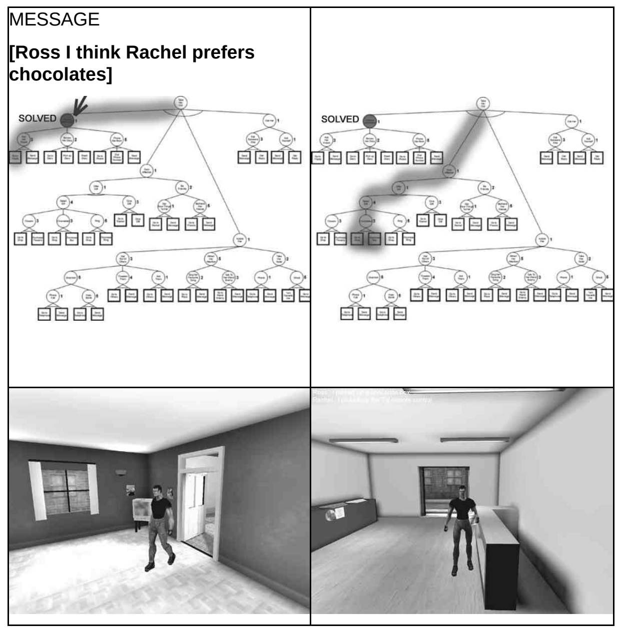

Figure 8. Providing information to characters.

Provision of information can also be used to trigger action repair. If for instance, Ross is looking for Rachel's diary and cannot find it at its default location, he can receive advice from the user ("the diary is in the other room") and repair the current action (this restores the executability condition of the read_diary action) (see Figure 8). In this case, spoken information competes with re-planning of another solution by Ross; The outcome will depend on the timing and duration of the various actions and of the user intervention (once a goal has been abandoned, it cannot, in our current implementation be restored by user advice).

Another form of linguistic interaction consists in giving advice to the characters. Advice is most often related to inter-character behaviour and social relationships. We have identified three kinds of advice. Generic advice is related to overall behaviour, e.g. "don't be rude". This can be matched to personality variables, which in turn determine the choice of actions in the HTN. Such advice can be interpreted by altering personality variables that match the heuristic functions attached to the candidate actions in the HTN. For instance, a "nice" Ross will refrain from a certain number of actions, such as reading personal diaries or mail, interrupting conversations or expelling other characters from the set. This of course relies on an a priori classification of actions in the HTN, which is based on static heuristic values being attached to nodes of the HTN.

Situational advice is a form of rule that should help the character avoiding certain situations. One such example is an advice to avoid making Rachel jealous, such as "don't let Rachel see you with Phoebe". The processing of such advice is more complex and we have only implemented simplified, procedural versions so far. One such example in the same situation consists in warning Ross that Rachel is approaching (Figure 9).

Figure 9. Advice "I think Rachel is coming".

Speech input mostly targets the plan-based performance of an actor's role but can also target other forms of behaviour as mentioned in section 4, such as situated reasoning or reactive behaviour. For instance, specific reactive behaviour can be inhibited by spoken instructions: Rachel can be advised not to be jealous ("Rachel, don't be jealous").

8. CONCLUSIONS

We have described a specific approach to interactive storytelling where the user, rather than being immersed in the story is essentially trying to influence it from his spectator position. We would suggest that this paradigm is worth exploring for future entertainment applications, where it could bridge the gap between traditional media and interactive media. The long-term interest of this approach is however a case for user evaluation, which should first require the system to reach a critical scale. Our prototype currently has four autonomous characters, all based on HTN plans (though the main character Ross has the most complex plan) and is able to generate short stories ("one-act plays", [18]) up to three minutes in duration, with approximately one "beat" [18] per minute. This contrasts with the objective suggested by Mateas and Stern [18] of 10-15 minute stories with three characters, which is certainly a valid objective for interactive storytelling systems. Performance of the planning component has shown good potential for scaling-up on simulated tests. The main difficulties are expected to arise from increased interaction between characters and the associated descriptions of situated reasoning, for which no clear methodological principles have been established. On the other hand, there is much to be learned from running larger-scale tests and these results could have a generic interest for the study of high-level behaviour of embodied characters.

9. REFERENCES

[1] 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.

[2] Swartout, W., Hill, R., Gratch, J., Johnson, W.L., Kyriakakis, C., LaBore, C., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J., Thiebaux, M., Tuch, L., Whitney, R. and Douglas, J., 2001. Toward the Holodeck: Integrating Graphics, Sound, Character and Story. Proceedings of the Autonomous Agents 2001 Conference.

[3] Aylett R., 1999. Narrative in Virtual Environments - Towards Emergent Narrative. In: AAAI 1999 Fall Symposium on Narrative Intelligence.

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

[5] Machado, I, Paiva, A. and Brna, P., 2001. Real Characters in Virtual Stories (Promoting Interactive Story-Creation Activities). First International Conference on Virtual Storytelling, Avignon, France. Lecture Notes in Computer Science n. 2197, pp. 127-134, Springer-Verlag.

[6] Bers, M. U. and Cassell, J., 1998. Interactive Storytelling Systems for Children: Using Technology to Explore Language and Identity. Journal of Interactive Learning Research 9(2) 183-215.

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

[8] Grabson and Braun, 2001. A Morphological Approach to Interactive Storytelling. Proceedings of Cast01, Living in Mixed Realities, pp. 337-340.

[9] Hayes-Roth, B., van Gent, R. and Huber,D., 1997. Acting in Character. In R. Trappl and P. Petta (Eds.), Creating Personalities for Synthetic Actors. Berlin, New York, Springer.

[10] Bickmore, T., and Cassell, J., 1999. Small Talk and Conversational Storytelling in Embodied Interface Agents. AAAI Fall Symposium, "Narrative Intelligence," November 5-7, Cape Cod, MA

[11] Mateas, M. and Stern, A., 2000. Socially Intelligent Agents: The Human in the Loop. AAAI Fall Symposium.

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

[13] Young, R.M., 2001. An Overview of the Mimesis Architecture: Integrating Narrative Control into a Gaming Environment. AAAI Spring Symposium in Artificial Intelligence and Interactive Entertainment, AAAI Press.

[14] Cavazza M, Charles, F. and Mead, S.J., 2001. AI-based Animation for Interactive Storytelling. Proceedings of Computer Animation, IEEE Computer Society Press.

[15] DeLeon, V., and Berry, R., 2000. Bringing VR to the Desktop: Are You Game?. IEEE Multimedia Magazine, April-June, Vol. 7, No. 2.

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

[17] Douglas, J. and Gratch, J, 2001. Adaptive Narrative: How Autonomous Agents, Hollywood and Multi-processing Operating Systems Can Live Happily Ever After. First International Conference on Virtual Storytelling, Avignon, France. Lecture Notes in Computer Science n. 2197, pp. 100-112, Springer-Verlag.

[18] Mateas, M. and Stern, A., 2000. Towards Integrating Plot and Character for Interactive Drama. In Socially Intelligent Agents: The Human in the Loop. Papers from the 2000 AAAI Fall Symposium, Technical Report FS-00-04, AAAI Press, 113-118.

[19] Webber, B., Badler, N., Di Eugenio, B., Geib, C., Levison, L., and Moore, M., 1995. Instructions, intentions and expectations. Artificial Intelligence J. 73, pp. 253-269.

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

[21] Kambhampati, S., Mali, A., and Srivastava, B, 1998. Hybrid planning for partially hierarchical domains. In Proceedings of AAAI-98.

[22] Kambhampati, S. and Hendler, J.A., 1992. A Validation Structure Based Theory of Plan Modification and Reuse, Artificial Intelligence, 55(2-3):193-258.

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

[24] Smith, S.J.J., Nau, D.S., and Throop, T., 1998. Computer Bridge: A Big Win for AI Planning. AI Magazine, 19(2):93-105.

[25] Weyhrauch, P., 1997. Guiding Interactive Drama. PhD Thesis, School of Computer Science, Carnegie Mellon University, Technical Report, CMU-CS-97-109, Pittsburgh, PA.

[26] 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.

[27] Cavazza M, Charles, F. and Mead, S.J., 2002. Emergent Situations in Interactive Storytelling. Proceedings of ACM Symposium on Applied Computing (ACM-SAC), to appear.

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

[29] Traum, D. and Rickel, J., 2001. Embodied Agents for Multi-party Dialogue in Immersive Virtual Worlds. IJCAI 2001 Workshop on Representation and Reasoning in Practical Dialogue Systems, Seattle, USA.

[30] Bindiganavale, R., Schuler, W, Allbeck, J.M., Badler, N.I., Joshi, A.K. and Palmer, M., 2000. Dynamically Altering Agent Behaviours Using Natural Language Instructions. Proceedings of Automomous Agents 2000, Barcelona, Spain.