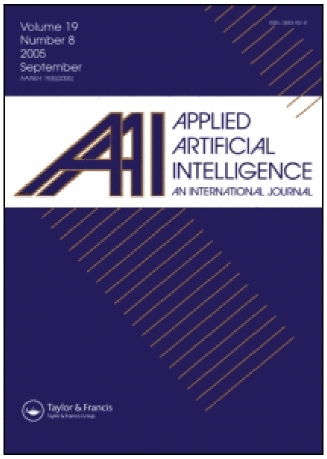

Applied Artificial Intelligence
An International Journal
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713191765

A COMPUTATIONAL MODEL OF AN INTELLIGENT NARRATOR FOR INTERACTIVE NARRATIVES

Nicolas Szilas [ab]
[a] TECFA-FAPSE, University of Geneva, Carouge, Switzerland
[b] Virtual Reality Lab, Department of Computing, Macquarie University, Sydney, Australia

Online Publication Date: 01 September 2007
To cite this Article: Szilas, Nicolas (2007) 'A COMPUTATIONAL MODEL OF AN INTELLIGENT NARRATOR FOR INTERACTIVE NARRATIVES', Applied Artificial Intelligence, 21:8, 753 - 801
To link to this article: DOI: 10.1080/08839510701526574
URL: http://dx.doi.org/10.1080/08839510701526574

Applied Artificial Intelligence, 21:753–801
Copyright © 2007 Taylor & Francis Group, LLC
ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839510701526574

A COMPUTATIONAL MODEL OF AN INTELLIGENT NARRATOR FOR INTERACTIVE NARRATIVES

Nicolas Szilas □ *TECFA-FAPSE, University of Geneva, Carouge, Switzerland and Virtual Reality Lab, Department of Computing, Macquarie University, Sydney, Australia*

□ *One goal of interactive narrative and drama is to create an experience while utilizing the computer where the user is one main character in the story. This goal raises a set of practical and theoretical challenges to artificial intelligence. In particular, an intelligent narrator has to dynamically maintain a satisfactory storyline and adapt to the user intervention.*

After reviewing the existing approaches to interactive drama, an original model is described, based on several theories on narrative. This model is based on a rule-based component for the generation of meaningful narrative actions and on a model of the user for the ranking of these actions and the action selection. A simple but authorable text generation system is also described, for the display of the actions on the computer.

The system is implemented on a real-size scenario and experimental results are discussed. We conclude by discussing the possibility of a wider application of our approach within the field of AI.

INTRODUCTION: TOWARDS INTERACTIVE DRAMA

Goal and Definition

This research aims at building systems and tools that enable the creation of a new kind of user experience called interactive drama.

Interactive drama is a narrative genre on computer where the user is one main character in the story and the other characters and events are automated through a program written by an author. Being a character means choosing all narrative actions of this character.

Before the proposal of an original approach to this problem, an overview of the narrative genres that approach interactive drama but fail to reach the criteria defined is proposed; then a review of related research projects on interactive drama is presented.

Address correspondence to Nicolas Szilas, TECFA-FAPSE, University of Geneva, Acacias 54, Carouge CH-1227, Switzerland. E-mail: Nicolas.Szilas@tecfa.unige.ch

The Apparent Paradox of Interactive Drama

From the beginning of the 1980s, various forms of narratives have been tried on computers in order to benefit from interactivity: hypertext literature, interactive fiction, and video games.

However, these artistic and/or entertainment pieces have highlighted shortfalls of interactive narrative: the interactivity on the story itself has always been limited. For example, players in an adventure video game would like to deeply modify the story through their actions, but they are always confined into a restricted path.

From the analysis of these works, we can bring out four main mechanisms used for building interactive narratives, possibly combined.

1. Discursive interactivity: The user modifies the ways s/he is given the story, but not the story itself. Narrative is usually divided into two parts: the story, which is the succession of chronological events which happen in the fictional world, and the plot/discourse, which is the succession of events as they are told to the audience, not necessarily in chronological order (Genette 1972; Eco 1979). This mechanism of interactivity typically occurs at the level of the discourse.
2. Superposed interactivity: The interactivity is inserted locally in the story, and the user's actions only have local influence. After the interactive session, the next predefined event in the story is triggered. Typically, the user finds a puzzle and has to solve it in order to reach the next event in the story. This local interactivity usually requires skill, either motor skill (fights in video game) or cognitive skill (puzzles) from the user.
3. Branching: The chain of events is replaced by a graph of events, each node corresponds to a choice for the user (Crawford 1982). If the graph is oriented, the author can control the various paths made available to the user. The main problem of this approach is that the author must design all possibilities, which grow exponentially with the number of nodes, in case of a tree structure. By cutting some branches (deadlocks) or merging some others, the number of possibilities is reduced but the user's choice does not have an impact on the main storyline. An improvement consists of using a memory of user's choice and using this information to modify the graph later in the story (conditional branching). It improves the user's feeling of having an impact, but it only constitutes a very partial solution to the problem of combinatorial explosion.
4. Destructured narrative: The graphs of events are not oriented, allowing various schemes of navigation between story events. In that case, except for very particular stories, the causality of the narrative is broken and the immersion inside the story world is not possible (Ryan 2001).

Looking at the set of video games that involve characters (the best candidates to be narratives), it is noticeable that the most interactive ones are real-time strategy games (Warcraft, Age of Empire) and simulation games (Theme Park, Creatures, The Sims^{TM}), but the user's actions are not meant to influence a story. Conversely, the most narrative games (adventure video games) are the least interactive.

Thus, we observe an inverse correlation between narrativity and interactivity (Crawford 1996; Aylett 1999; Szilas 1999; Louchart and Aylett 2003). A story is a chain of events that has been well crafted by an author, while interactivity seems to imply that the user will destroy this authored order and write his/her own story with the elements of narrative provided by the author.

Is this inverse correlation a fundamental principle of interactive narrative, as advocated in Juul (2001), or does it depend on the shortage of existing works produced so far? This research aims at finding new mechanisms for combining interactivity and narrativity, in order to invalidate the inverse correlation mentioned previously and open the way to a new kind of narrative artistic form.

The main applications of this research are interactive entertainment (including video games), interactive art, pedagogical virtual environment, and semi-automatic screenwriting.

Origins of the Field

The interest of artificial intelligence (AI) for narrative is not new. During the 1970s and the 1980s, several researchers attempted to generate stories with various AI techniques (e.g., Lehnert 1981; Wilensky 1983; see Lang 1999 for a review). Some effort was spent to find general formal definition of stories, which happened to be impossible (Bringsjord 1999).

Some story generation systems use a narrative engine whose principles could be used for interactive drama. However, they do not provide a solution to interactive drama. Interactive drama raises new questions concerning choice and influence. In particular, the key issue is to manage the quantity of choice at a given point in the narrative, as well as the influence of those choices on the narrative.

Current projects on interactive narrative and AI stem from the research on virtual environments developed in the 1990s. Indeed, virtual environments populated with virtual characters are potential grounds for developing narratives.

The Oz Project at Carnegie Mellon University (Bates 1992) initiated the field of interactive drama, by exploring the potential of virtual environments and intelligent agents for interactive narrative. It developed the notions of "broad and shallow" agents (Bates et al. 1992), containing a large spectrum of abilities (cognitive and emotional) in order to build believable rather than realistic agents. This project opened the way to AI applied to arts, where the user's involvement in the AI simulation is more important than the proper mechanisms used within the system.

There are a growing number of projects for interactive stories and drama currently in development (Young 1999; Szilas 1999; Magerko 2003; Spierling et al. 2002; Stern and Mateas 2003; Louchart and Aylett 2004). There are a number of new conferences exploring the topic (AAAI Symposium on Narrative Intelligence, COSIGN–Conference on Semiotics for Games and New Media, International Conference on Virtual Storytelling, and TIDSE-Technologies for Interactive Digital Storytelling and Entertainment). These denote a growing interest on interactive narrative and entertainment, in general.

Basic Properties of a System for Interactive Narrative

If AI is to be used for telling stories, does it mean that AI will replace human story tellers? Does it mean that AI must be creative?

In interactive narrative systems, the author is still present and plays a central role. Interactive narrative is neither about replacing the human storyteller, nor assisting it (except for a few specific applications). It is about creating a new experience, where a user interacts with the elements of a story.

Authors should be able to bring all their creativity into the system, in order to provide the user with a good interactive experience. In other words, the participation of the user does not steal some creative power from him. On the contrary, it demands more effort to manage this participation.

An interactive narrative contains three components: the author, the computer, and the user. Situated between the user and the author, the computer, endowed with AI capabilities, has two main functions.

- Provide an enjoyable interactive narrative experience to the user.
- Allow the author to write the content.

Some researchers focus on the first function (Stern and Mateas 2003), while others focus on the second one and build authoring tools (Spierling et al. 2002). In all cases, any interactive narrative system must encompass both the writer and the reader points of view.

OVERVIEW OF APPROACHES FOR MODELING THE NARRATIVE

About Models for Narrative

Any attempt to build a system for interactive narrative is guided by a set of theories about what a narrative is. These theories can yield a computer model of narrative.

There are plenty of different theories of narrative, coming from very diverse fields such as literature, linguistics, anthropology, cognitive psychology, psychoanalysis, dramaturgy, screenwriting, aesthetics, etc. Furthermore, most theories of narrative are not scientifically established, so the truth of one theory against another one is controversial itself. Finally, whether a text is narrative or not varies between different disciplines and people. Without any unified model of narrative, we have decided to build our own theory of narrative, which takes some elements from various existing theories. No claim is made that this theory is better than others, but it aims at satisfying the two following principles.

- It covers enough features for the purpose of interactive narrative.
- It is suitable for computation.

This theoretical model will be described next, followed by the computer model which is derived from it. In the following sections, we provide general principles that served as guidelines for the development of the narrative model.

Fine Grain/Coarse Grain Models

A computational model of interactive narrative consists of deconstructing a story into elementary pieces and letting the computer reassemble the pieces according to the user's actions. The authoring contains two main parts.

- Writing the elementary pieces.
- Defining the rules to combine the pieces.

The larger the size of these pieces, the easier the authoring, but it has an inverse effect on interaction. Indeed, a coarse grain generally corresponds to a small scene in a story, which has its own interest and can be rather intuitively authored. But story-level interaction has to occur mainly between the pieces, so the larger the grain, the more limited the interactivity.

For example, the system described in Spierling et al. (2002) uses the Propp model (Propp 1928), which is a decomposition of a story into 31 elementary basic functions, such as "violation" or "trickery." This coarse grain model reduces the interaction either between the functions, through a oriented graph, or inside each piece. But the latter interaction has relatively small influence on the global story (superposed interactivity).

The system called Façade (Mateas and Stern 2000; Stern and Mateas 2003) is a compromise: A story is described as a composition of *beats*, which are micro scenes. The grain is finer than many models based on scenes, but still coarse enough to ensure authorability of each piece. Furthermore, the beats are variable, interruptible, and interwoven, which creates variability and interactivity. However, the interactivity remains mostly local: The user's action triggers a beat and modifies how the beat happens, but it does not have a precise influence on the content of future beats. This limitation is compensated by the choice of the story itself, which allows a loose ordering of the beats it is composed of.

To build a system where the user has a greater and more precise influence on the story, one needs a fine grain model, which decomposes the story into basic actions.

The Erasmatron system (Crawford 1999) is such a system. The drama comes from the succession of actions calculated by the system and selected by the user; each action is a sentence, based on one *verb*. These verbs, by themselves, are loosely narrative: "beat up," "flatter," "insult," etc., but the system intends to create interesting narrative sequences by the dynamic arrangement of these actions.

Intelligent Agents Models for Narrative and Their Limitations

Characters are the main components in drama writing (Egri 1946; Lavandier 1997). Thus, one solution for interactive drama apparently consists of simulating those characters with intelligent agents. The interaction of rich virtual characters with cognitive, social, and emotional intelligence would produce interesting narrative, where the user would be one of those characters. The video game industry is evolving in that direction, adding more and more intelligent abilities into its characters (see games like Black and White, The Sims^{TM}, Half-Life 2$^{®}$).

The task of building intelligent characters is difficult because it involves a full modeling of humans, following a "broad and shallow" approach. It is also different from traditional AI, because the accuracy of the modeling of humans is less important than its effect on the users (Sengers 1998); realism is less important than believability.

Despite its appeal, and beyond the technical difficulties, there is however a fundamental limitation in this approach. A narrative is not the result of the simulation of psychologically motivated characters but it is a construction, where each action is designed according to the whole. The notion of function introduced by Propp more than 75 years ago (Propp 1928) demonstrates this point: A function is a character action *defined from the intrigue viewpoint* (Szilas 1999). Gérard Genette studied the motivation of a given action inside a narrative and concludes that the motivation of characters is used to dissimulate the narrative function (Genette 1969). Thus, the narrative works with a kind of inverse causality: Characters behave also according to what *will* happen in the story ("the determination of the means by its ends [...], of the causes by its consequences," Genette [1969] p. 94, our translation). This is true for any kind of narrative.

A variation of the intelligent character approach is the intelligent *actor* approach. In that case, an agent behaves towards an audience. It can combine psychological-based actions and narrative-based actions. The best example of this ability in real life is theatrical improvisation, where several actors, hardly briefed during a few seconds, can play a small drama together. In the system developed at the University of Teesside, short soap-like dramas are generated through the interaction of intelligent actors. They are actors and not characters in the sense that their plans include potentially interesting actions (Cavazza et al. 2001). But no real acting strategy, which would include narrative constraints, has been implemented so far. As a result, only a subset of the generated stories are interesting. Another approach for synthetic actors uses the notion of role (Donikian and Portugal 2004; Klesen et al. 2001). Inspired by the theory of Greimas (1970), the system developed in Klesen et al. (2001) uses two characters who take two roles, respectively: the helper of the protagonist and the helper of the antagonist. By balancing their intervention according to the narrative situation, these characters contribute to create a permanent conflict for the user. However, this strategy remains rather limited and specific, in comparison to the rich catalog of intervention "given" to a real improvisation actor.

So far, implementations of improvisational strategies to create interactive drama have remained insufficient because these strategies are complex. As coined in Louchart and Aylett (2002), these strategies would consist of a "distributed game master," but from a computational point of view, distributing the game master or drama manager does not necessarily make the task of building interactive drama easier.

Related to the exclusive use of autonomous agents is the notion of narrative emergence, introduced in Aylett (1999). In addition, in other domains of AI, a complex global phenomenon emerges from the interaction of simple units; the narrative could emerge from the local interaction of intelligent agents. The concept is quite seducing, but unfortunately, one still does not know what are the conditions for a narrative to emerge.

These limits of the use of intelligent agents for interactive drama often lead some researchers to propose some mixed architectures, with a centralized unit controlling the narrative and several units corresponding to characters. The centralized unit, often called the drama manager, is omniscient and gives instruction to the virtual actors.

The concept of drama manager is typical in interactive drama architectures, and was already introduced in the Oz Project (Kelso et al. 1992). But it was initially designed for sporadic intervention, the agents being responsible of most of their behavior. However, the difficulties to build agents that integrate narrative constraints suggests that the role of the drama manager should be much greater, influencing every high-level action of agents. A similar conclusion has been the basis of the development of Façade (Mateas and Stern 2000): The agents carry out the management of movements, emotions, and behaviors, but do not make any high-level decision.

Our own research focuses on the drama manager, leaving out, temporarily, the topic of the agent's behavior.

The Simulation of Narrative

A narrative is organized into a sequence of events. What does interacting with a sequence mean?

A promising approach considers that the sequence is a result of the simulation of a system. Interacting with a narrative is then equivalent to interacting with a system that has been designed to produce stories.

To clarify the concept, let us take a metaphor. Suppose that you want to build a virtual pinball on a computer. The ball is performing some complex trajectories, according to the obstacles it bumps. To make it possible for a user to modify these trajectories, starting from the trajectories themselves would be inefficient. On the contrary, you have to simulate the mechanical laws that govern the ball displacements. Then interacting with a pinball is easy, because you do not interact with the trajectories, but with mechanical elements used in the simulation. Similarly, to efficiently design an interactive storytelling system, we promote an approach where a narrative is expressed in terms of narrative laws, and where the computer simulates those laws in real time. The main challenge is to find a useful set of laws. It does not mean finding the universal laws of narrative, which would cover any narrative (which seems to be impossible, according to Bringsjord and Ferrucci [1999]), but to find some laws which cover enough varieties of narrative to make the interaction rich and interesting.

A key property of those laws is that they are based on an atemporal model: A model for simulation is always composed of static elements, which produce temporal phenomena by the process of simulation. Back to our pinball analogy, the rules for calculating the trajectories handle stiffness, inertia, angle, past position of the ball, etc., which are atemporal elements. The mechanics-based rules do not handle pieces of trajectories, like straight lines or parabolas. This atemporality has been a strong guideline in choosing one theory of narrative rather than another in the design of a computer model of narrative.

Existing Interactive Narrative Systems Based on Atemporal Models

The two following systems are closely related to the system proposed here, because they are based on atemporal models of narrative.

In the mid 1990s, N. Sgouros used AI to dynamically generate plots according to users' actions. Given a set of goals, motives, and norms for characters, the system calculates actions that create interesting dramatic situations and properly resolve all the plots and subplots. Based on Aristotelician theory (Aristotle-330), the interactive drama uses conflicts between characters' goals and norms (Sgouros 1999).

Planning techniques are well suited to interactive narrative, because they can reproduce how the audience makes expectations while perceiving a narrative. Since 1995, the Liquid Narrative Group at North Carolina State University has been developing an approach of interactive narrative based on planning, which intervenes at several levels: as a tool for adapting the story to the user's interaction (Young 1999; Riedl et al. 2003), for the camera control (Jhala and Young 2005), and as a way to model suspense in narrative (Somanchi 2003).

The model proposed in the next section intends to extend these works by allowing more interactivity and by increasing the number of narrative features managed by the engine.

PROPOSITION OF A MODEL OF NARRATIVE

The Theoretical Model

The model introduced here is a model of narrative that combines several findings in narratology. Indeed, any particular study on narrative is limited, focusing on one aspect of narrative and neglecting the others. The following theoretical model is the basis for the computer model developed in the next section.

We consider that any narrative, and in particular any drama, contains three dimensions. Each dimension is necessary to build a narrative.

The first dimension is the discourse dimension. It means that any narrative is a discourse: it aims to convey a message to its reader/viewer/listener (Adam 1994). This pragmatic view of narrative puts the narrative at the same level as any rhetorical form, even if narrative conveys its message in a very particular manner. As stated in Adam (1994), the characters' actions serve as a pretext to convey this message. This view of narrative is also adopted by Michael Young for interactive narrative (Young 2002), and allows him to make an analogy between dialog systems and interactive narrative systems.

In particular, the discourse dimension conveys a kind of morality. A narrative is not neutral: It conveys the idea that some actions are good, some are not. The good and bad evaluation is relative. It depends on a system of values, implicitly used by the author (Jouve 2001). Note that this system of values can conform to the global cultural values of the audience or in the contrary, can be inverted: The narrative can then be considered as "subversive," since it conveys an ideology that is contrary to the "culturally correct" morality (Jouve 2001).

As an example, in a fable written by Aesop, the morality is usually included in the text itself: "Slow but steady wins the race." In most texts, although not visible, the morality is implicitly present.

If the discourse dimension is omitted, the audience asks: "so what?" (Adam 1994, p. 108), and the narrative is a failure.

The second dimension is the story dimension. A narrative is a specific type of discourse, which involves a story. The story is described as a succession of events and character actions, following certain rules. Structuralism has proposed several logical formulations for the narrative (Todorov 1970; Greimas 1970, Bremond 1974). At the heart of these models is the concept of narrative sequence, initiated by an initial imbalance, and terminated by a final evaluation or sanction (Todorov 1970). The story itself is a complex combination of several sequences. Note that although Propp's model (Propp 1928) contains a unique sequence, Bremond demonstrated that even Russian folktales, which served as an experimental corpus for Propp, consist of several sequences superimposed (Bremond 1974). The story level is also described in the theory and practice of screenwriting (Vale 1973; Fields 1984; McKee 1997).

The third dimension is the perception dimension. Indeed, we found that the detailed models of narrative proposed by structuralism are incomplete (Szilas 2002).

- How is a single sequence temporally organized regarding the duration of time, beyond the ordering of its elements?
- How do the sequences work together?
- Why does one sequence follow a certain route versus another?

It seems that the study of how the narrative is perceived during reading/viewing/listening is the answer to these questions. Within perception, a key factor is emotion.

Barthes wrote in (1968): "le récit ne tient que par la distorsion et l'irradiation de ses unités" ("the narrative holds because of the distortion and the irradiation of its units"), and then discussed the notion of suspense in terms of an emotion. We have here one example of emotion, but there is not one single emotion in narrative. Another component of drama is conflict (Field 1984; Egri 1946; Lavandier 1997), which produces a highly emotional response to the viewer. In films, in particular, the narrative consists of building elaborated cognitive representations of the story in the viewer's mind, in order to produce various emotions such as fear, hope, frustration, relief, etc. (Tan 1996).

Generally speaking, the detailed role of emotion is often neglected in narratology. According to Noel Carroll, "What is not studied in any fine-grained way is how works engage emotion of the audience" (Carroll 2001 p. 215). Emotions are not a nice component to have of narrative, but are a "condition of comprehension and following the work" (Carroll 2001). More precisely, according to Noel Carroll, emotions play a central role for focusing the attention of the audience.

In fiction, the audience experiences a large spectrum of emotions, as though the depicted events were real. There are various theories that explain why the readers or viewers have these emotions although they know that the events do not exist (Schneider 2006). The reader or viewer creates an internal representation of these events which share features with real events. According to the pretend theory, the audience pretends that the events are real and feels "quasi-emotions" (Schneider 2006). According to the thought theory (Carroll 2001), imagining an emotional situation is enough to provoke an emotion (aren't we disgusted by thinking of a disgusting thing?). According to the illusion theory (Tan 1996, p. 236), the viewers consider the fictional events as almost real, as if they were inside the fictional world. Despite their diversity, all these approaches confirm the cognitive nature of emotions in narrative. Emotion is controlled by the narrative via the manipulation of two dimensions.

- The cognitive structure: How the user builds an internal representation of the fictional world, including the beliefs, desires and intentions of characters, and possible future fictional worlds.
- The mode of presentation: How the fictional events are related to real events, bringing up issues such as realism, immersion, etc.

So far, this research on narrative intelligence has focused mainly on the first dimension to manage the emotional component of narrative.

Nonemotional perceptive factors also play a role inside the narrative. For example, it is classical in drama to consider that some actions allow to characterize some characters. The choice of such an action is not guided by the story itself but by the fact that the audience must know the characters in order to understand the story.

If the perception dimension is omitted, it would give a "syntactically correct" narrative, but the audience would neither understand it clearly nor get engaged in it.

The Architecture

We are building a system, called IDtension, which is an implementation of the principles discussed previously. Its general architecture (see Figure 1), can be divided into five modules.

1. The World of the Story: It contains basic entities in the story (characters, goals, tasks, subtasks or segments, obstacles), as written by the author, as well as all the facts describing the current state of the world. Most of these facts are deduced by the system. They include the states of the characters (defined with predicates) and facts concerning the material situation of the fictional world (the fact that a door is closed, for example).
2. The Narrative Logic: From the data stored in the world of the story, it calculates the set of all the possible actions of the characters.
3. The Narrative Sequencer: It filters the actions calculated by the narrative logic in order to rank them from the most interesting to the least interesting. For this purpose, the sequencer uses the model of the user.
4. The Model of the User: It contains the estimated state of the user at a given moment in the narrative. It provides to the narrative sequencer an estimation of impact of each possible action on the user, with a set of narrative effects.
5. The Theatre: It is responsible for displaying the action(s), and it manages the interaction between the computer and the user.

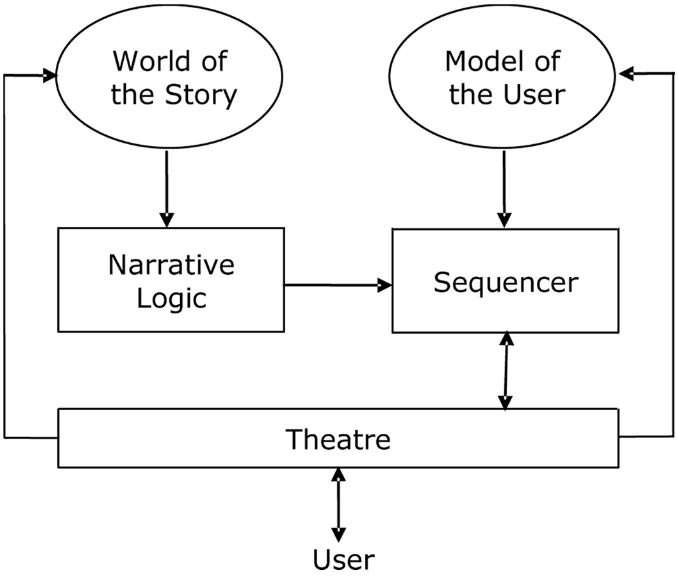

FIGURE 1 General architecture of the system (see text).

These modules interact with each other as follows: The user, by interacting with the theatre, chooses an action, which is sent to the other modules in the system. Then, the narrative sequencer receives the set of all possible actions from the narrative logic. It filters out the actions to be performed by the user and calls the model of the user to rank the remaining actions from the most satisfactory to the least satisfactory, with a scoring mechanism. The narrative sequencer then chooses one action among the most satisfactory ones, with a random factor. Then this action is sent to the Theatre to be displayed. In this interaction scheme, the user and the computer play one action in turn, but this simple choice is temporary: The computer may play several actions in one turn, as well as the user.

Note that if all characters are controlled by the computer, then the system can be turned into a story generation system (Szilas 2003).

The Narrative Logic

Actions

The narrative logic calculates all possible characters' actions, at a given point in the story. An *action* is defined as an elementary character's act. Actions are divided into performing acts (physical modifications of the world), speech acts (in dialogs), and cognitive acts (internal decisions of characters). The various types of actions currently in use are listed in Table 1.

The choice of the types of action described in Table 1 has been strongly guided by narrative theories (Todorov 1970; Greimas 1970; Bremond 1974; Adam 1994); see Szilas (2002) for a discussion of the structuralist roots of the IDtension system. The types of action described in Table 1 generate a typical elementary narrative sequence, composed of information of performance, influences, decisions, performance itself, and sanctions. Thus, the set of actions aims to be very generic, covering a large range of existing and possible stories.

The main feature of the action definition represented in Table 1 is that two notions are dissociated: actions and tasks. A task is author-defined and describes a possible act from a character, in a given story. An action is what is effectively performed during the storytelling, taking the task as an argument. The cross between the types of actions described in Table 1 and the tasks created by an author generates a multitude of actions, due to the combinations. This has two advantages.

1. From the author point of view: By creating one task, many possible actions are automatically generated. Without the dissociation, the author should redefine a task like "dissuade," its parameters, its role according to values, etc. In fact, such a complex authoring is not feasible. With the dissociation between tasks and actions, complex actions are generated,

TABLE 1 Different Types of Actions Handled by the Narrative Logic

Type of action	Category	Meaning	Main consequence (see Table 2)
Decide(X,g,par)	cognitive act	*X* decides to reach goal *g* with parameters *par*, henceforward noted *(g,par)*	WISH(X,g,par)
Inform(X,Y,F)	speech act	*X* informs *Y* about a fact *F*	KN0W(Y,F)
Encourage(X, Y, t, par)	speech act	*X* encourages *Y* to perform the task *t* with parameters *par*, henceforward noted *(t,par)*	
Disssuade(X,Y,t,par)	speech act	*X* dissaudes *Y* to perform the task *(t,par)*	
Accept(X,t,par)	cognitive act	*X* accepts to perform *(t,par)*	WANT(X,t,par)
Refuse(X,t,par)	cognitive act	*X* refuses to perform *(t,par)*	
Perform(X,t,par)	performing act	*X* performs the task *(t,par)*	
Congratulate(X,Y,t,par)	speech act	*X* congratulates *Y* for having performed *(t,par)*	
Condemn(X,Y,t,pa r)	speech act	*X* condemns *Y* for having performed *(t,par)*	
Ask_for_assistance(X,Y,g,par)	speech act	*X* asks *Y* to perform goal *(g,par)* for him/her.	
Accept_to_assist(X,Y,g,par)	speech act	*X* informs *Y* that s/he accepts to assist him/her by taking care of the goal *(g,par)*	WISH(Y,g,par')-The parameters might be different from original parameters.
Refuse_to_assist(X,Y,g,par)	speech act	*X* informs *Y* that s/he refuses to assist him/her by taking care of the goal *(g,par)*	

and open the way to interesting generated dialogs in the story, not only physical actions, as in most video games.

2. From the user point of view: The user perceives this two-level structure as a richer potentiality for action. For example, s/he understands that s/he can inform any character about the fact that s/he wants to perform a task; s/he also understands that s/he can influence (encourage or dissuade) any other character, as soon as s/he is aware that this character could perform any task, etc.

This feature distinguishes our system from other AI-based narrative systems (Young 1999; Cavazza 2001).

Data Stored in the World of the Story

The world of the story (see Figure 1) contains the following types of elements.

- Physical entities (characters, material objects, places)
- Goals: Some states in the world of the story that characters want to reach; when a goal is reached, facts are added in the world, or suppressed from it.
- Tasks: Acts that can be performed by the characters (see details next)
- Obstacles: Practical elements in the world of the story that make some tasks fail
- Attributes: Properties that serve as argument of various facts
- Facts: Predicates taking elements as arguments

The goals and tasks are parameterized. For example, the goal "possess an object" contains as a parameter the object that is desired. This parameter is instantiated when a character decides to reach the goal through the predicate "WISH": *WISH(X,g,par)* means that character *X* wishes to reach goal *g*, with the list of parameters *par*.

The same structure is used with tasks. The parameters of the task are instantiated when the character has the possibility to perform the task, through the predicate "CAN:" *CAN(X,t,par)* means that character *X* has the possibility to undertake task *t*, with parameters *par*. Note that the parameters of tasks are not necessarily the same as the parameters of goals: The task "steal" takes two parameters, the desired object and the victim of the theft, while the corresponding goal, "possess an object" only takes one parameter, the desired object.

The tasks themselves are decomposed into a linear succession of segments. Performing a task is equivalent to performing the current segment. Each segment contains:

- Pre-conditions: The condition to be met in order that the segment can be performed

- Consequences: Facts that are added to the world or suppressed from it

While physical entities, goals, tasks, and obstacles define the static elements in the world of the story, as described by an author, facts describe the current relationship between these elements. Table 2 summarizes the types of facts currently used in the system. Note that the term "know" is used instead of the widely used "believe." Since no uncertainty is managed at the level of beliefs, the two terms are similar. The term "know" has been chosen because it is more explicit for non AI people (screenwriters, game designers, etc.) and because it is used in narratology.

The Rules

The narrative logic stores rules, which calculate which actions are possible, at a given moment in the story. The left part of each rule contains preconditions describing a certain state of the world and the right part contains one action.

The preconditions are composed of two types of elements: *facts* (see previous section) and *constraints,* which are functions defining relations between its arguments. During the unification process, the constraints are used only once all arguments have been instantiated. Constraints allow to handle complex preconditions without having to define the generic predicates associated to these predictions.

The rules are all fired in parallel, providing the set of possible actions at a given moment in the story. Table 3 exhibits three examples of rules, among the current set of about 50 rules.

TABLE 2 Main Types of Facts Stored in the World of the Story and Handled by the Narrative Logic

Predicate Form	Description
WISH(X,g,par)	*X* wishes to reach goal *(g,par)*
CAN(X,t,par)	*X* is able to perform task *(t,par)*
WANT(X,t,par)	*X* wants to perform task *(t,par)*
HAVE_BEGUN(X,t,par)	*X* has begun to perform task *(t,par)*
HAVE_FINISHED(X,t,par)	*X* has finished to perform task *(t,par)*
HAVE_BEEN_BLOCKED(X,t,par,o)	*X* has been blocked by obstacle o while to perform task *(t,par)*
KNOW(X,F)	*X* knows a fact F
HINDER(o,CAN(X,t,par)	Obstacle *o* hinders *X* to perform the task *(t,par)*
CAUSE(E,o)	If the set of facts *E* is verified, then the obstacle *o* has higher risk to been triggered
HAVE(X,m)	*X* has the materiel object *m*
IS(X,p)	*X* is affected with the property *p*
IS LOCATED(X,1)	*X* is located in place *l*
ALLOWS_TO_REACH(t,g)	The task *t* allows to reach the goal *g*. This predicate explicitly makes the link between tasks ad goals

TABLE 3 Three Examples of Rules Used in the Narrative Logic

Name	Preconditions (Left Part)	Triggered Action (Right Part)	Description

Ending information	HAVE_FINISHED(X,t,par)
	KNOW(Z,HAVE_FINISHED(X ,t,par))
	~KNOW(Y,HAVE_FINISHED(X,t,par))
	different(X,Y)
	different(X,Z)
		inform(Z,Y, HAVE_FINISHED(X,t,par))	If a character *Z* knows that another character *X* has finished to perform a task, then he could inform a third character *Y* about this fact.

Encouragement	CAN(X,t,par)
	KNOW(X,CAN(X,t,par))
	KNOW(Y,CAN(X,t,par))
	~KNOW(Y,WANT(X,t,par))
	~KNOW(Y,HAVE_BEGUN(X,t,par))
	~KNOW(Y,HAVE_FINISHED (X,t,par))
	different(X,Y)
		encourage(Y,X,t,par)	If a character *X* is able to perform a task and knows it, if another character *Y* knows about this possibility but does not know that *X* has either started or finished the task, then the latter character could encourage the former to perform the task.

Condemnation	KNOW(Y,HAVE_FINISHED(X,t,par))
	different(X,Y)
		condemn(Y,X,t,par)	If a character knows that another character has finished a task, he can condemn him/her for this.

The Obstacle Modeling

An obstacle corresponds to a practical situation within the world of the story which hinders some tasks to succeed. Obstacles are a key component of narrative, allowing to distort the narrative sequence, according to Barthes (1968). The theories of screenwriting insist on the major role of obstacles (Vale 1973; Fields 1984). Note that the term "external conflict" is also used in that domain. An obstacle is characterized by its natural probability, or risk, to be triggered.

For example, an author can decide that the obstacle "slip on the floor" has a natural probability of 0.1. But if "it is raining," the probability can be set to 0.5. The fact that "it is raining" is called the *cause* of the obstacle.

In this example, there is not much to do about the cause, except to want for it to spontaneously disappear. Let us take another example: the obstacle "the car breaks down" could have as a cause "the car is not well maintained." In that case, the cause can trigger a new goal, namely, "go to the garage." Thus, obstacles are used as a subgoaling mechanism.

The different levels of knowledge about obstacles provide various narrative situations. For example,

- If a character ignores that there is an obstacle, its occurrence is a surprise.
- If a cause exists and if s/he knows it, she can try to change it, which could trigger a new goal.
- If s/he does not know the existence of the cause, hearing about it is an interesting new development in the narrative.
- If s/he knows that there is an obstacle and that the risk is low, then s/he can perform the task, with a limited suspense. But if the risk is high, and s/he has no choice but to perform the task, then the suspense is higher (Carroll 2001, p 227).
- If s/he knows the cause and s/he cannot modify it (for example, time or weather), s/he has to choose between performing the task anyway or wait for a change of condition, or abandon the task.

In our implementation, an obstacle is an autonomous object, which is associated to a segment of task. The same obstacle can be reused in different segments. This object contains the following traits.

- Parameters: Similarly to tasks and goals, obstacles are parameterized
- Cause: A list of conditions which, when satisfied, set the higher risk
- Risk+: Probability of triggering, if the cause is true
- Risk−: Probability of triggering, if the cause is false

- Characters_knowing_the_obstacle: Characters who know the obstacle (can be defined generically, with obstacle parameters or with the generic parameter "actor")
- Characters_knowing_the_cause: Characters who know which triggers the obstacle, the *cause* mentioned previously (which does not mean that they know if it is true or not)
- Consequences: Facts added to the world of the story if the obstacle is triggered

Two predicates are used to allow exchange of information around obstacles: *HINDER* and *CAUSE*. A character *X*, who can perform task *t* (with parameters *par*), can have various knowledge, represented by facts in the world of the story.

- *KNOW(X,HINDER(o,CAN(X,t,par))* means that *X* knows that the obstacle *o* hinders him to perform task *t* (with parameters *par*).
- *KNOW(X,CAUSE(E,o))* means that *X* knows that condition *E* causes the higher level of risk of occurrence, concerning obstacle *o*.
- *KNOW(X,E)* means that X knows condition *E*, which appears to cause an obstacle.

Depending on the existence of this knowledge, several dramatic situations are produced around one obstacle.

The character's perception of the risk is different, depending on the character's knowledge. The model contains a perceived risk, which evolves in time. For example, some incentives or dissuasions can modify the perceived risk.

As a result, three types of risks of triggering can be distinguished.

- The natural risk, a probability defined by the author
- The perceived risk, a probability linked to a character and updated during the story
- The effective risk, the probability chosen for the performance of the task. It is derived from the natural risk but is also biased by narrative factors.

The Management of Delegations

The fact that characters can help each other is very common in narrative. Narratology has identified six fundamentals roles in narrative; one of them is the helper or adjuvant (Greimas 1970; Souriau 1950 in Elam 1980).

A specific way for characters to help each other consists of delegations: One character who wishes to reach a goal can ask another character to reach this goal for him/her.

The delegation is achieved by three specific types of actions: *Ask_for_assistance, Accept_to_assist,* and *Refuse_to_assist* (see Table 1). Three corresponding rules are used in the narrative logic.

The motivation of a character to ask for assistance (estimated in the model of the user) can be a quite complex phenomenon. Among the various motivations are the fact that the other has more chance to succeed, the fact that the character has too many tasks to perform in parallel, the fact that the other will have less negative consequence, the wish not to take risk and/or suffer from negative consequences (egoism), a subordination relation between characters, etc. We have limited the complexity to an assistance motivated by the presence of one or several obstacles in the known tasks reaching the goal. In particular, the chances to ask for assistance are higher if:

- the risk of the less risky known task is high, and
- the risk of the corresponding obstacle for the potential helper is low.

Similarly, the motivation for accepting of assistance has been reduced to two factors:

- the sympathy between the two characters, and
- the interest of the potential helper for the initial goal.

Depending on the nature of the goal being delegated, the delegation mechanisms are slightly different. Consider the three following goals (with parameters not yet instantiated).

a) destroyed(object)
b) owned(actor, object)
c) strong(actor)

In those examples, "actor" is a parameter that means the character who wishes the initial goal. In the first case, the delegation does not make any problem. In the second case, the parameter "actor" is naturally replaced during the delegation. Thus, when the delegated goal is reached, the initial goal is not yet reached: The helper owns the object but the initial character does not. In that case, we define a *completion goal,* undertaken by the helper, which aims at completing the delegation. In this example, the completion goal is "have_transmitted(actor,consignee,object)." In the third case, the delegation is simply not possible.

Global Representation of the World of the Story

Figure 2 provides a visual representation of the various components defined in the world of the story: Characters (circles) wish to reach some goals (squares). Each goal can be reached through tasks (arrows) that are more or less negatively evaluated, according to each value of the narrative (oblique lines). The characters are more or less linked to the values (bold and dashed lines). Obstacles allow the triggering of a subgoal (through the validation of X and Y). {+X} (respectively, {−X}) denotes the fact that the goal adds (respectively withdraws) X to the world of the story when it is reached. In the example depicted in Figure 2, a new goal can be decided by the character because the withdrawn condition of the second goal is the fact that causes an obstacle.

This visual representation of the world of the story plays a central role in authoring. This is an atemporal description of a story, as it has to be built by a writer, see Szilas et al. (2003) for a discussion on authoring.

Action Selection

The System of Values

In the model, a value, or ethical value, is a numerical scale between −1 and 1, where tasks are placed. A task is said to be attached to a value when the task is placed in the negative part of the corresponding scale. A task is said to be neutral according to a given value when it is placed at zero, on the scale corresponding to the value. The positive part of the scale is not used currently. Note that such a definition of a value comes from

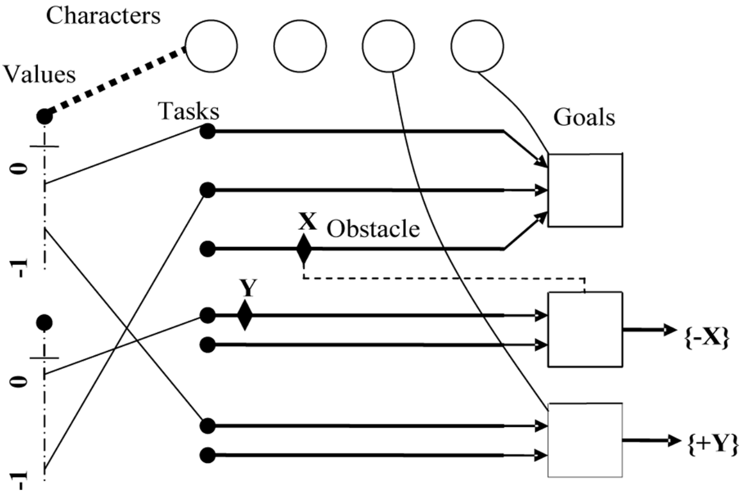

FIGURE 2 Goal task structure.

narratology and is totally different from the value as defined in some manuals of screenwriting (McKee 1997). Values are used:

- to assess the conflict associated with various tasks,
- to assess the propensity of a character to perform a task,
- to assess the propensity of a character to encourage/dissuade another character about a possible task, and
- to assess the propensity of a character to congratulate/condemn another character for having performed a task.

The two last items show that an attachment to a value is not just a traditional character's feature: it not only modifies the character's own performances, but also the way the characters judge each other. For example, "good memory" is a feature: You would usually not condemn someone for having done something against this feature. But "honesty" is a value, because you would condemn a dishonest person.

There are several values in a given narrative, but their number should be limited: the fact that several tasks are attached to a small set of values creates a unity in the narrative, and enables to convey the system of values from various points of view.

The Model of the User

The narrative logic provides a large set of actions that are possible to play at a given moment in the narrative. As described previously, these actions have been designed to be potentially meaningful in a story, in the sense that they catch some basic properties defined in various narrative theories. However, these properties do not constitute a sufficient condition for narrative. In other words, the narrative logic overgenerates: It produces sequences that are not considered as stories.

To constrain the produced sequence more, one needs a function to assess each possible action regarding the quality of the story it creates. This assessment should be based on a model of how the user would perceive the possible action. This part of the system is called the model of the user. The model of the user is able to assess the interest of each proposed action, by giving a score between -1 and 1. It is up to the module called the sequencer to either choose the maximum scored action or randomly choose amongst the highest scored actions (as it is the case in the current implementation).

This score is the combination of several subscores, each one corresponding to a *narrative effect*. This allows a modular design. Defining a definitive scoring function is a difficult task, since narrative theories only provide hints towards that goal. Having several criteria allows to experimentally test each criterion and vary their relative impact in the global function at the time of testing. Note that the combination of the effects is not linear.

We use the term "model of the user" in reference to the "model of the reader," as defined by Umberto Eco (1979), and not the term "user model," as classically used in the human computer interaction literature. The reason is that the term "user model" classically refers to the model of the specific characteristics of the end user of a system, like the age, skills, or tastes. Here, the model of the user is different: It is a generic model of a user, which simulates a general user who gathers data specific to the story told so far; it corresponds to the user that a narrator has in mind for predicting the impact of each event. Such a model of the user can even be used in noninteractive story generation (Baileys 1999).

The Narrative Effects

As stated before, there is no established list of necessary and sufficient narrative effects. Here follows a tentative list that has been implemented.

The Ethical Consistency. It is central that characters in a story behave in a consistent way. If, at some point, they behave in a way showing that they are attached to some ethical values, they do not have to change their attitude later in a story (unless the character evolves during the story, which is another issue).

For actions directly related to a task *(encourage, dissuade, accept, refuse, perform, congratulate, condemn)*, and in the case of one unique value, the ethical consistency is the product of the attachment of the actor to the ethical value by the position of the task according to that value. In case of multiple values, the arithmetic mean is calculated.

Formally, let us define:

- A, the action to be scored;
- p_A, the character who undertakes the action;
- t_A, the task related to this action; and
- par_A, the parameters of the task t_A.

For example, if A is "encourage(Paul, Joe, Steal, [key, Ann])," which means "Paul encourages Joe to steal the key from Ann," then p_A is "Paul," t_A is "Steal," par_A is "[key, Ann]."

Given an ethical value v, two basic variables are known.

- $pos(t_A, par_A, v)$, the position of the task t_A and its parameters par_A, according to the value v, as defined by the author (between 0 and 1)
- $att(p_A, v)$, the attachment of the character to p_A the value v, as defined by the author or filled dynamically (see next for this latter option)

- Then, given V the number of values in the story, the ethical consistency is

$$EC(A) = \sigma_A \times \frac{1}{V}\sum_v post(t_A, par_A, v) \times att(p_A, v),$$

where σ_A denotes the *strength* of action A, a constant depending on the type of the action A, comprised between -1 and 1. An encouragement, for example, would have a strength of 0.9, while a dissuasion would have a strength of -0.9. This means that it would be consistent for a character to dissuade another character to perform a task s/he estimates bad (both the attachment and the strength are negative, which gives a positive ethical consistency), but it would be inconsistent for this character to encourage such an action (the strength is positive this time).

For all other actions the ethical consistency is set to zero (except for *inform HINDER*, a case that is not detailed here).

In the current implementation, the attachments of characters to values are fixed at the beginning of the story. But in the future, it might be interesting to explore the idea of letting this attachment evolve according to the user action, by employing an adaptation process. Thus, to which values each character is attached would be defined dynamically, to better fit the user's own values (Szilas 2001).

The Motivation. It seems obvious that characters should preferentially choose the actions that give them the most chance to reach their goal. This is carried out by the criterion called *motivation.* Behind this simple definition lies, in fact, a vast problem, which can be viewed as the general problem of AI, if AI is considered through a rational agent approach: "Acting rationally means acting so as to achieve one's goal, given one's belief" (Russel and Norvig 2003, p. 7). More precisely, assessing the motivation of agents for an action A might require to calculate all possible outcomes of the action associated with their probability of occurrence, from the agent point of view. Such an approach is not suited to our goal, for the following reasons.

- The characters should not be "over intelligent," by producing complex inferences that would then appear unrealistic and not believable.
- It is not worth producing a complex reasoning from a character if the user is not able to understand this reasoning. Explanation mechanisms could be provided, but the complexity of the language generation associated with such a mechanism is far too high.
- The environment where the character evolves is complex and uncertain: It is complex because of the large amount of actions that are possible due to a combinatorial effect. It is uncertain because all action include a

random component, in particular, the triggering of obstacles. Thus, the problem would soon become intractable.

- Following the "broad and shallow" principle for believable agents (Bates et al. 1992), which is a fortiori true for interactive drama, there is no need for the most intelligent agents but for the most interesting ones. Our effort should not be grabbed by a particular point in the system, but should be dispatched on all of its aspects.

Thus, we have designed ad hoc formulas to calculate the motivation. Contrary to the ethical consistency, the formula is often specific for each type of action.

Three main components, ranged between 0 and 1, are used to evaluate the motivation.

The interest: It measures the degree to which the actor of the action is interested in the achievement of the goal targeted by the concerned task. The calculus of the interest takes into account various factors, such as the nature of the goal linked to the action (whether it is an initial goal or a sub-goal), the character who holds the goal (which might not be the actor of the action evaluated), and the various encouragements concerning this goal that the actor has received.

For actions that are directly linked to a task (see previous list), let us define:

- ag_A, the agent that would perform the task linked to the action A,
- G_A, the goal linked to the action A (it corresponds to the "WISH" predicate),
- *importance (p,G)*, the importance of the goal G for the character p, as defined by the author; and
- *perceivedImportance(p,G)*, the importance of the goal G for the character p, taking into account the various influences that p has received: It is increased/decreased when p is encouraged/dissuaded to perform a task leading to G.

The following algorithm is used to determine the interest of the actor p_A for the goal G_A.

```
BEGIN
    IF $p_A = ag_A$
        $interest(p_A, G_A) \leftarrow perceivedImportance(p_A, G_A)$
    ELSE
        $interest(p_A, G_A) \leftarrow Importance(p_A, G_A)$
    IF IsSubgoal($G_A$)
    $interest(p_A, G_A) \leftarrow \lambda \times interest(p_A, G_A) + (1-\lambda) \times interest(p_A, supergoal(G_A))$
END,
```

where λ is a constant in [0, 1] weighting the contribution of the subgoal's interest.

This simple recursive algorithm means that the author writes the importance of initial goals for each character, and that the system derives the interest for subgoals.

The perceived risk: To each obstacle known by a character is associated a perceived risk that represents the risk that the obstacle will be triggered, according to this character.

Let us denote *pRisk(X,o)* the perceived risk of character *X* regarding the obstacle *o*. This value is created whenever *KNOW(X,HINDER(o,CAN(X,t, par)))* is added to the world of the story. It is initialized as follows.

- When the knowledge of the obstacle comes from another character Y:
- $pRisk(X, o) \leftarrow pRisk(Y, o)$
- Otherwise:

```
IF NOT $\exists E/CAUSE(E, o)$
    $pRisk(X, o) \leftarrow risk^{+}(o)$
ELSE IF KNOW(X,CAUSE(E,o)) AND KNOW(X,E)
    $pRisk(X, o) \leftarrow risk^{+}(o)$
ELSE IF KNOW(X,CAUSE(E,o)) AND KNOW(X, ~E)
    $pRisk(X, o) \leftarrow risk^{-}(o)$
ELSE
    $pRisk(X, o) \leftarrow \frac{risk^{+}(o)+risk^{-}(o)}{2}$
```

Then, during the story, the perceived risk is updated according to two mechanisms.

- When a knowledge is added concerning either a CAUSE predicate or the truth of the cause itself (the fact E in formulas), then the perceived risk is updated according to formulas.
- When the character fails to perform a task, its perceived risk is incremented.

$$pRisk(X, o) \leftarrow pRisk(X, o) + RISK\_INCREASE\_RATE \times (1 - pRisk(X, o)),$$

where RISK_INCREASE_RATE is in [0,1].

This latter update means that the characters will perceive a task more risky if they fail to perform it.

The strength: As with the ethical consistency, the strength allows to weight the motivation according to the type of action, as well as invert the effect between positive types of actions (encourage, accept, perform, and congratulate) and negatives ones (dissuade, refuse, and condemn).

Finally, one can give the motivation for each type of action.

- For *encourage* and *dissuade*:

$$MOT(A) = \sigma_A \times interest(P_A, G_A) \times \Phi\left(\prod_{o\in L(p_A,t_A,par_A)} (1 - pRisk(p_A, o))\right),$$

where:

- $L(p_A,\ t_A,\ par_A)$ is the list of all obstacles on the task (t_A, par_A) known by character p_A.
- Φ is a correcting function, which sets a minimal value even if the risk is 1, so that characters influence tasks even if they are risky.
- For *accept*:

$$MOT(A){=}\sigma_A{\times}interest(P_A,G_A){\times}\Phi\left(\prod_{o\in L(p_A,t_A,par_A)} (1{-}pRisk(p_A,o))\right){\times}comp(t_a),$$

where $comp(t_A)$ is a correcting factor, which lowers the motivation when another competing task is wanted by the character, to reach the same goal. This makes the characters focus on one task at a time without always jumping between tasks for the same goal.

- *For Perform*:

$$MOT(A) = \sigma_A \times interest(P_A, G_A) \times \Phi\left(1 - \prod_{o\in L(p_A,t_A,par_A,S)} (pRisk(p_A, o))\right)$$
$$\times\ wait(t_A),$$

where:

- s is the currently performed segment.
- $L(P_A, t_A, par_A, s)$ is the list of all obstacles on the task (t_A, par_A), for the segment s known by character p_A.
- $wait(t_A)$ is a correcting factor, which lowers the motivation when a subgoal of the task is currently being worked towards. The task is then in a "wait" mode.
- For *condemn* and *congratulate*:

$$MOT(A) = \sigma_A \times interest(par_A, G_A)$$

The motivation of *refuse* and several cases of *information* is set to zero. It means that these actions are considered as neutral, from the motivation point of view. However, for special cases of information, the motivation is calculated specifically.

- When a character informs about an obstacle (HINDER) concerning the addressee, either to dissuade him to perform the task or in contrary to warn him about a danger.
- When a character informs about his intention to perform a task (WANT), but that task concerns a goal against the addressee (in that case, the motivation should be negative)
- When a character informs about a possibility for the addressee to perform a task (CAN), but the task concerns a goal of negative importance to him (in that case, the motivation should be negative)

These cases will not be detailed here, neither the motivation of *decide* nor the actions related to the delegations.

The Relevance. In such a fine-grained decomposition of narrative, the sequence of actions might be rather uncoordinated. For example, *X* could ask a question to *Y*, then *Y* asks a question to *X*, then *Y* answers to *X*, and then *X* answers to *Y*! To avoid such situations, we have implemented a relevance criterion, which is inspired from the relevance maxim, as introduced by Grice: “Say things related to the current topic of the conversation.” A similar criterion has been used in (Weyhrauch 1997) the *thought flow*, which “measures whether one event in the User’s experience relates logically to the next.”

Again, instead of defining the relations between events specifically for each story, it is necessary to define generic relations, in order to alleviate the author’s work.

Inspired from the work of the narratologist C. Bremond (1974), we introduce in the model the notion of “process,” which is a pair of two actions. A process contains one initiating generic action, and one or more terminating generic actions. Seventeen of such generic processes have been created (see Table 4 for examples of processes).

During the execution, the processes are instantiated into *sequences.* Like a process, a sequence is composed of an initiating action and one or more terminating actions. The sequences are managed by the model of the user, which processes each action played by the system and seen by the user as follows.

- It tries to match (unify) the action to the left part of each process. If a match occurs, then a new sequence is created, which instantiates the matched process.

TABLE 4 [Examples of Three Processes, with State Evolution and Corresponding Actions] p and q Denote Characters, g Denotes a Goal, t Denotes a Task, Par Denotes the Parameters of the Tasks or Goals. The States Column Illustrates the Evolution of Facts During the Process. The Actions Column Contains One Initiating Action, and One or More Terminating Actions

Description	States	Actions

After a character *p* informs *q* that s/he wishes to reach a goal, the character *q* answers by providing a mean to reach the goal.	∼KNOW(q,WISH(p,g,par))
	KNOW(q,WISH(p,g,par))
	KNOW(p,CAN(p,t,pa))
		Inform(p,q,WISH(p,g,par))
		Inform(q,p,CAN(p,t,par))

After a character *p* informs *q* that s/he could perform a task *t*, the character *q* answers by either encouraging or dissuading him/her to perform the task or informing about an obstacle on the task.	∼KNOW(q,CAN(p,t,par))
	KNOW(q,CAN(p,t,par))
		Inform(p,q,CAN(p , t, par))
		Encourage(q,p,t,par) or
		Dissuade(q,p,t,par) or
		Inform(q,p, HINDER(obs,CAN(p,t,par)))

After a character *p* informs *q* that s/he has finished a task, the character *q* condemns or congratulates *p* concerning the task.	∼KNOW(q, HAVE_FINISHED(p,t,par))
	KNOW(q, HAVE_FINISHED(p,t,par))
		Inform(p,q, HAVE_FINISHED(p,g,par))
		Condemn(q,p,t,par) or
		Congratulate(q,p,t,par)

- It compares the action to the actions of the right part of the existing sequences. When they are the same, the *sequence* is deleted.

To calculate the relevance, one just needs to evaluate if the action *A* terminates a sequence that has been opened recently.

The algorithm for calculating *REL(A)*, the relevance of A, is as follows.

```
BEGIN
    $D_A \leftarrow \{sequences\ s / terminates(A, s)\}$
    IF $D_A = \emptyset$ THEN $REL(A) = 0$
    ELSE
    $recency_A \leftarrow \min_{s \in D_A}\{currentTime - time(initialAction(s))\}$
    $REL(A) \leftarrow max\left(1 - \frac{recency_A - 1}{MAX\_RECENCY}, 0\right)$
END
```

This algorithm allows the relevance to be continuous: The more recent the previous action that the action is relevant to, the more relevant the action. *MAX_RECENCY* is a narrative parameter that denotes the delay from which the relevance is zero. The time measurement is not the real time but the index of each action. Thus, *MAX_RECENCY* is an integer, typically between 1 and 3.

Suitable Complexity. One basic feature of narrative is to create some expectations in the minds of the audience. For example, as soon as the audience knows that a character can perform a task, it will start raising the question: "Will the character perform that task or not?," until the character performs the task or explicitly abandon it. Expectations in a narrative are multiple, which creates a certain level of complexity. A sufficient complexity is needed, in order to keep audience's interest high. But too much complexity will finally get the audience lost. In the beginning of the narrative, the complexity is null, since the audience is not raising any specific question. At the end of the narrative, it is usually the case that many questions raised during the narrative are answered, thus the complexity is low. (Some narratives, on the contrary, do leave some open questions, but we will not consider this case, for the sake of simplicity.) So, an ideal curve of complexity emerges: starting at zero, reaching an average level, and then decreasing at a low level.

How do we quantify the complexity within our formalization of narrative? The same structures which serve for the relevance, namely the processes, can be used. The number of sequences in the model of the user gives a measure of complexity. To define the suitable complexity associated to an action *A*, let us define:

- O_A, the number of sequences opened by action *A*,
- C_A, the number of sequences closed by action *A*,

- *N*, the number of sequences in the model of the user, and
- *desiredComplexity*, the number of open sequences that is desired.

One can then calculate:

- deficit $= \varphi(\text{desiredComplexity} - \text{N})$, which measures in what extent complexity should be increased (deficit > 0) or reduced (deficit < 0), and
- $\text{complexification}_{\text{A}} = \varphi(\text{O}_{\text{A}} - \text{C}_{\text{A}})$, which measure in that extent the action A complexifies the story ($\text{complexification}_{\text{A}} > 0$) or simplifies it ($\text{complexification}_{\text{A}} < 0$),

where φ is a normalization function which keep values between -1 and 1. φ is defined as follows.

$$\varphi(x) = \frac{min(M, max(-M, x))}{M},$$

where *M*, a positive integer constant, is the maximum deficit or complexification considered.

The suitable complexity is then given by

$$SC(A) = deficit \times (1 - |deficit - complexification_A|)$$

This formula allows that the most suitable complexity is reached when the deficit equals to complexification.

Note that the way of calculating *desiredComplexity* has not yet been specified. It is initially set at a fixed author-defined value. After a while, *desiredComplexity* should be set to zero.

Progression. We found it useful to add a criterion to favor actions that make the story go forward, rather than actions that make the story stagnate. The estimation of this criterion is based on the type or subtype of action involved (see Table 5). Furthermore, a repetitive action is badly rated, according to the progression criterion. The final algorithm is:

```
BEGIN
    IF A has been played within the last REPETITION_WINDOW visible actions
        progression ← REPETITIVE_ACTION_PROGRESSION
    ELSE
        progression ← Progression(A)    (see Table 5)
END
```

This criterion allows for an author to easily bias towards a certain type of actions.

TABLE 5 Progression Constants

Type of action	Progression
Decide	0.7
Inform (general)	0.5
Inform CAN (towards agent)	0.9
Inform HINDER (towards agent)	0.9
Inform HAVE BEEN BLOCKED	0.7
Encourage	0.1
Disssuade	0.1
Accept	0.7
Refuse	0.1
Perform	0.7
Congratulate	0.6
Condemn	0.5
Ask_for_assistance	0.5
Accept_to_assist	0.6
Refuse_to_assist	0.6

Conflict. Conflict is the most important narrative effect in the model of the user, not only because conflict is viewed as the core of drama, but also because conflict makes the link between the values the author wants to convey in general and the particular emotional expression of this value. The type of conflict we are interested in is the internal ethical conflict. It occurs when a character has to perform a task that violates his/her values.

Conflict is also an effect highly dependent from the context of the occurring action, as we will see next.

The suggested principles to create conflict are as follows.

- The very challenge of interactive drama is to put the user himself/herself into a conflicting situation. Thus, the conflict generated inside the user's character is the main focus.
- The expression of conflict should be based both on the short-term level, that is the immediate conflictual nature of an action, but also on a strategic level: how an action can promote conflict later in the story.
- The strategic level of conflict management is based on the fact that a goal should not be reached without conflict, if conflicting tasks exist to reach that goal.
- The level of conflict in interactive drama will not be as high as for noninteractive drama. One cannot push the user towards really uncomfortable situations; otherwise, the user will quit the story and remain frustrated not to be able to complete the story the way s/he wanted (Ryan 2001). Thus, if enough conflict has been expressed in a given situation (according to the author's tuning), even if the user has

not chosen to accomplish the conflictual task, the system should "free" the user and provide a nonconflictual way to reach the goal.

In order to implement these various levels of conflict management, it is first necessary to calculate the conflict related to a task *t*, with parameters *par*, performed by the character *p* to reach goal *G*. This conflict is linked to the balance between two components, the motivational component, denoted *m* and the ethical component, denoted *e*.

$$\mathrm{m(p, t, par, G) = interest(p, G)}$$

$$\mathrm{e(p, t, par, G)} = -\mathrm{Min}\{\mathrm{pos(t, par, v)} \times \mathrm{att(p, v)}\}$$

Conflict is maximal when *m* and *e* are both positive, *m* equals *e*, and when *m* and *e* are large, as depicted in Figure 3. This is produced by the following formula.

$$\begin{aligned}\mathrm{conflict(p, t, par, G)} =& (1 - |\mathrm{m(p, t, par, G)} - \mathrm{e(p, t, par, G)}|) \\ & \times \min(\mathrm{m(p, t, par, G), e(p, t, par, G)}) \\ & \quad \text{if} \quad \mathrm{m(p, t, par, G)} > 0 \text{ and } \mathrm{e(p, t, par, G)} > 0 \\ & 0 \quad \text{otherwise}\end{aligned}$$

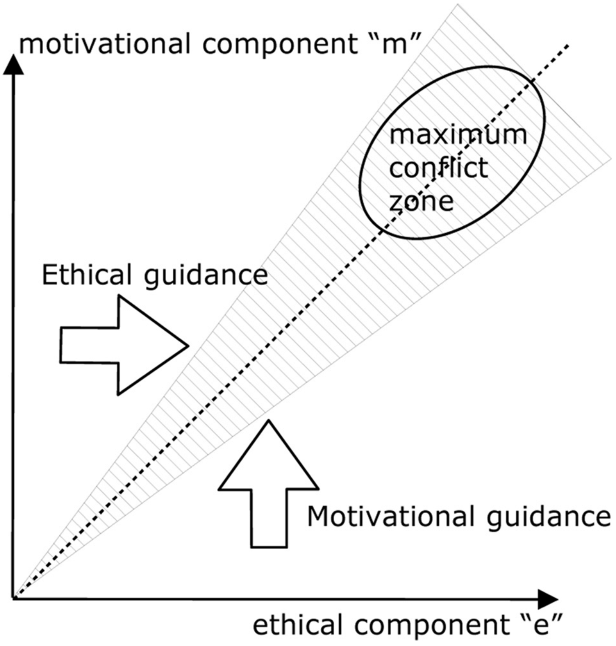

FIGURE 3 Balancing the conflict via guidance.

Second, one needs to estimate if enough conflict has been expressed for a given goal, at any moment in the story. A Boolean variable called *conflicting(G)* is set to TRUE if conflict can still be expressed around this goal, FALSE otherwise. *conflicting(G)* is calculated as follows.

```
BEGIN
   IF expressedConflict(G) < potentialConflict(G)
   conflicting(G) ← TRUE;
ELSE
   conflicting(G) ← FALSE;
END
```

where:

- *expressedConflict(G)* is an integer variable initialized at 0 and incremented each time an action containing a conflicting task pointing to *G* is played by or towards the user.
- *potentialConflict(G)* is an integer initialized when the goal *G* is created at the product of the number of conflicting tasks attached to *G* by the number of conflicting actions that should be displayed by conflicting task (an author-defined constant).

The formulas to calculate the conflict are different depending on the type of action.

1. For *encourage* towards the user:

The first component of the conflict, named motivational guidance, consists of balancing the motivational and the ethical components of conflict by pushing the user towards the task. As depicted in Figure 3, conflict is maximal when *e* and *m* are balanced, and when their value is high. Guidance, either ethical or motivational, consists in pushing the user towards the diagonal dashed lines of the graphic, to maximize the conflict. Such guidance occurs only when the task is outside the hatched zone. The algorithm for *encourage* is as follows.

```
BEGIN
   IF conflicting(GA)
      AND   $\mathrm{conflict}(\mathrm{ag}_A, \mathrm{t}_A, \mathrm{par}_A, \mathrm{G}_A) > 0$
      AND
         $\frac{\mathrm{e}(\mathrm{ag}_A,\mathrm{t}_A,\mathrm{par}_A,\mathrm{G}_A)}{\mathrm{m}(\mathrm{ag}_A,\mathrm{t}_A,\mathrm{par}_A,\mathrm{G}_A)} > \mathrm{THRESHOLD\_RATIO\_MOT\_ETH}$
   THEN
      motGuidance(A) ← GUIDANCE_CONTRIBUTION
   ELSE
      motGuidance (A) ← 0
END
```

where THRESHOLD_RATIO_MOT_ETH and GUIDANCE_CONTRIBUTION are two constants:

- THRESHOLD_RATIO_MOT_ETH delimits a zone around the balancing line (see Figure 3): outside this zone, a balancing component is added
- GUIDANCE_CONTRIBUTION is a fixed value, in the interval [0,1]

The conflict component is finally given by:

$$C(A) = \frac{\text{motGuidance}(A) + \text{conflict}(p_A, t_A, par_A, G_A)}{2}$$

2. For dissuade:

Following the same idea as for encourage, a dissuasion can pull a user from performing a task by stressing the ethical point of view, and thus balance the motivational and ethical components (see Figure 3, ethical guidance).

```
BEGIN
   IF conflicting(GA)
      AND $\text{conflict}(ag_A, t_A, par_A, G_A) > 0$
      AND $e(ag_A, t_A, par_A, G_A)/m(ag_A, t_A, par_A, G_A) >$ THRESHOLD_RATIO_MOT_ETH
      ethGuidance (A) ← GUIDANCE_CONTRIBUTION
   ELSE
      ethGuidance (A) ← 0
END
```

Finally:

$$C(A) = \frac{\text{ethGuidance}(A) + \text{conflict}(p_A, t_A, par_A, G_A)}{2}$$

3. For *accept/refuse/perform* from the user:

$$C(A) = \text{conflict}(ag_A, t_A, par_A, G_A)$$

4. For *felicitate, condemn* towards the user:

$$C(A) = \frac{\text{conflict}(ag_A, t_A, par_A, G_A) + \text{conflict}(p_A, t_A, par_A, G_A)}{2}$$

This formula means that the conflict is due to the conflictual nature of the task, according to both the agent (which in that case is the user) and the actor (the character who felicitates/condemns).

For all other cases, except special types of information that will not be detailed here, the conflict is set to 0.

Action Display (the Theatre)

This section describes how the actions played by the user and the actions played by the nonplayer characters are presented to the user.

Which Media?

The architecture is designed to be medium-independent: the world of the story, the narrative logic, the sequencer, and the model of the user generate a narrative in an abstract way, which can be then displayed on any medium.

The medium that is primarily targeted for the system is a three-dimensional, real-time environment, as in many popular video games.

The implementation described in this report is text-based. Actions are described textually, rather than being visually represented in the virtual world. Such text-based displays require generating dynamic text, according to the action calculated by the engine.

This language generation component is also necessary in a 3D world: the characters need it in order to be able to produce speech. We have conducted experiments in that direction, by using the conversational agents developed in Pelachaud and Bilvi (2003) as part of the theatre (Szilas and Mancini 2005). Also, an integration of the narrative engine described here with a commercial game engine is currently under development. Only the text component will be described here.

Constraints on Text Generation

The text generation module has to transform a structure like inform(X,Y,WANT(Z,t,par)), for the parameters: X = joe, Y = mary, Z = paul, t = steal, par = [ring,charles] into a sentence like: Joe to Mary: “Did you know that? Paul wants to steal the ring from Charles.”

Techniques for transforming an abstract content into a lexical forms do exist, in the field of natural language generation, or NLG (Reiter and Dale 1997). These techniques involve the use of discursive, grammatical, and lexical knowledge in order to transform the abstract representation into a text.

Our goal is however different from other more classical applications: AI is not used as a powerful way to solve a problem but as a creation tool. It aims to be used by an author or artist to create an interactive experience. Such an author is neither a linguist nor a programmer. With current linguistically oriented NLG systems, such an author could not easily modify the textual content, and the system would use a standard language level (standard English for example). This is suitable for systems dedicated to train reservations or information retrieval, but not acceptable for artistically authored systems such as interactive drama.

Another technique to generate text is called "template-based generation" (Reiter 1995; Buseman and Horacek 1998; Gedolf 1999; Theune 2000). It consists of assembling pieces of canned text containing gaps to be filled according to the context. This approach is better suited to our application, because the author can create these pieces of text more easily and freely. For example, s/he could use slangs or even nongrammatical sentences if s/he desires that a character has to use that level of language.

Template-based techniques are inherently less powerful than NLG techniques because they are less maintainable and do not allow as much variability as NLG techniques (Reiter 1995). However, they are more and more used in nontrivial text generation (Theune 2000; Stone and DeCarlo 2003) and can be effectively hybridized with NLG techniques, blurring the distinction between the two families of techniques (Deemter et al. 2001).

A Flat Template-Based Approach for Text Generation

To each atomic element in the system is associated a piece of text written by an author. This association is simply entered into a spreadsheet. To generate the sentence given previously, several lines should be entered, as illustrated in Table 6.

This example shows that the text generation is recursive: A template calls another template, which calls another template, etc.

Things are more complex when pronouns are needed. Suppose one wants to express the same abstract sentence inform(X,Y,wants(Z,t,par)), with the parameters: X = Joe, Y = Mary, Z = Paul, t = steal, par = [ring, Mary] (Mary is the victim of the theft instead of Charles). This should give a sentence like: Joe to Mary: "Did you know that? Paul wants to steal the ring from *you.*"

TABLE 6 List of Necessary Templates to Generate the Sentence: Joe to Mary: "Did You Know That? Paul Wants to Steal the Ring from Charles."

Story Element	Template
inform	[actor] to [addressee]: "Did you know that? [content]."
want	[agent] wants to [task:infinitive]
steal (infinitive)	steal [desired_object] from [owner]
joe	Joe
mary	Mary
paul	Paul
charles	Charles
ring	the ring

TABLE 7 Templates Associated to One Task, with All Its Forms

Story Element	Form	Template
steal	infinitive	steal [desired_object] from [owner]
steal	infinitive_owner_speaker	steal [desired_object] from me
steal	infinitive_owner_addressee	steal [desired_object] from you
steal	infinitive_owner_player	steal [desired_object] from you

In that case, "you" has to be used, instead of "Mary." Thus we need to introduce several templates for one story element, depending on the relation between the parameters. To each story element is thus associated several forms, as illustrated in Table 7. In this example, the same story element is associated to four different forms, according to a possible match between the parameter of the task (here "owner") and one of the persons involved in the communicative act (the speaker and the addressee). For example, the second form, named "infinitive_owner_speaker" is the case where the owner and the speaker are the same person.

Finally, we have introduced some possible variants for each template.

- some variants depending on the style (usually attached to a character)
- some variants to be randomly chosen.

An example of the variants is finally given in Table 8.

Note that part of the templates are reusable across stories, the one that concerns the types of action and states. But an author might prefer adapting this part to his or her own will.

Future Improvements of NLG

As can be observed from the example given previously, the price to pay for the ease of writing one single template is the multiplicity of the templates, for one story element (see Table 7). This is obvious when the list of parameters involves more than one character, as in the task: "ask_to_influence" (ask *someone* to influence *somebody*). In that case, there are $2^3 = 8$ different variants, regarding the parameter's matching.

TABLE 8 Different Templates for a Given (Element, Form) Couple

Story Element	Form	Style	Template
steal	infinitive	neutral	steal [desired_object] from [owner]
			snoop [desired_object] from [owner]
		crude	jack [desired_object] from [owner]
		polite	purloin [desired_object] from [owner]

Furthermore, not only task but also goals and obstacles should be processed this way.

We reach here the limit of the pure template-based approach. Future work will consist of adding some linguistic processing to get an hybrid template-based system (Theune 2000). Our main challenge at this stage is to keep the templates easy to author. Various solutions are investigated.

Currently, each action is displayed independently, without lexical connections with each other. Anaphora processing for pronouns is needed to avoid repeating the whole name of each person or object ("he," "it"), as well as anaphoric expression ("the guy"). This task is part of the "referring expression generation" task in NLG (Reiter and Dale 1997).

At last, transitions between actions could be improved by introducing variants informed by the narrative engine. For example, if, inside a dialog, the character changes the topic, a small text like "By the way," could be added.

EXPERIMENTAL RESULTS

Method

Testing an interactive drama system is a difficult task because it involves, at the end, user testing. Given the ergonomic limitation in the input and output interfaces, one needs to find another way to test our engine.

A two-part evaluation is suggested.

- *Narrativity*: From a given scenario, we will show one example of a completed interactive session, when the story reaches its end. The end is set up manually when the main goal is reached. The narrative properties of the resulting story will be discussed, taking into account its interactive nature. Indeed, the narrative interest of an interactive drama does not lie in the narrative interest of the story produced by the system and read a posteriori. This will be a qualitative evaluation.
- *Choice*: For this session, we measure the number of choices provided to the user at each turn. By calculating the mean of these values over all turns, the quantity of interactivity can be globally estimated. We measure as well the number of *new* choices, with respect to the previous turn, which is the number of actions proposed to the user which were not proposed in the previous turn (this value is usually positive but can be negative when there are more options that disappear than options that appear between two turns). The mean of this value tells how much the story evolves and provides new opportunities to the user. This is an estimation of the interest of the interactive drama.

The Scenario

The choice of a scenario is determinant in the results. It was found that within our formalism, completely different levels of narrativity and interactivity could be obtained, depending on the authored structures, not mentioning the artistic quality of the scenario itself.

For example, the first scenario that has been implemented, even if it demonstrated some features of the system, was too simple to allow complex variations in the storyline (Szilas 2003). The current scenario is more complex and possesses a key feature for interactivity, namely circularity, or recursivity (Szilas and Rety 2004). It means, in the simplest case, that a goal *A* is the subgoal of a goal *B* and that goal *B*, with different parameters, is the subgoal of goal *A*: Goals recursively call each other, creating diversity with a limited initial material.

The scenario, co-written with Olivier Marty, has the following setting: "It is mid-17th century. A Dutch galleon is navigating back to the Old Continent from the Carribeans. The transportation of slaves having been successful, the captain, Nelson decides to return home much earlier than scheduled. On the deck stands a large cage initially used to incarcerate rebel slaves. In lieu of slaves, four haggard and depressed looking men made prisoners as a result of a failed attempted plunder by an English vessel a few days before. Among them, there is Rak, a freed slave who is dying to see his native land. Next to him, stands a so-called Lord seemingly lost in his thoughts. He is wearing a laced, dirty costume apparently ripped from the battle resulting from the plunder. A former sailor, known as Jack, has a single idea in mind: planning a riot which will help him get rid of Nelson, the captain. Finally, there is Malcolm, a young French boy whose frail physique seems quite inadequate to undertake the trip under such conditions. The crew comprises Menil, Bello, and Nelson, the captain who is accompanied by Clara, his supposed niece." The user plays "Jack."

The scenario contains 7 goals, 15 tasks, and 19 obstacles. Three goal-task structures out of 7 are displayed in Figure 4. Other goals are "Know taste," "Improve speaking abilities," "transmit object" (useful in case of delegation of the goal "Possess"), and "transmit knowledge" (useful in case of delegation of the goal "Know taste"). In this scenario, there is a main goal (getting rid of the ship's captain), but most actions occur at the level of the other goals. The textual expressions have been written separately, as explained previously. They will not be detailed here.

Results

Narrative Analysis

The Appendix displays the story, as seen by the user (without manual English editing). This example shows that the story reached completion,

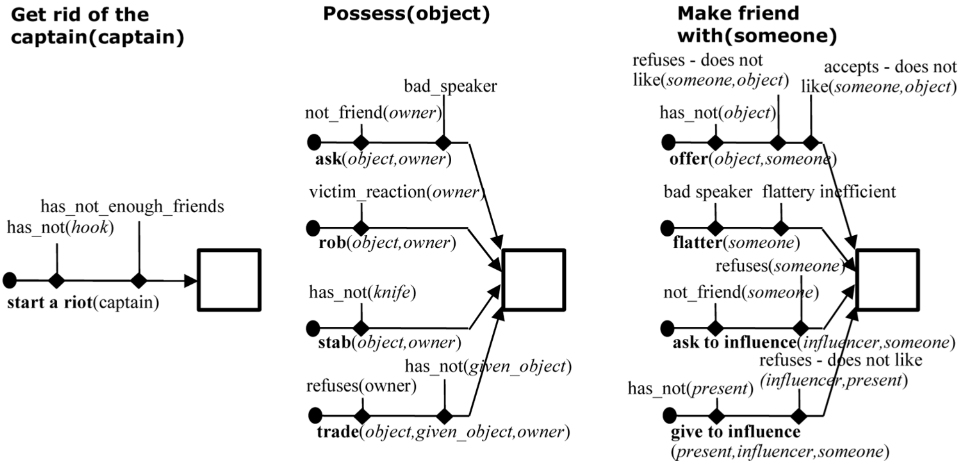

FIGURE 4 Extract of the goal task structures for the scenario. Note that the values are not represented, for the sake of readability.

which is never obvious with such a fine-grain decomposition of the narrative. Note that only actions seen by the user are displayed. In particular, other actions between other characters occurred but were not seen by the user. These actions can just be inferred by him or her.

The story exhibits several interesting narrative situations, such as:

- The main goal being established before the story starts, several subgoals were triggered: possessing the hook (#6), having Malcolm as a friend (#15), talking better (#19) and possessing the book of rhetoric (#25).
- These goals are linked together causally, creating a cohesion in the story.
- Several obstacles generate interesting complications in the story (see actions #4, #11, #17, #18, #26).
- The protagonist controlled by the user could delegate a goal (possess the hook) to another character (Lord), who got the hook from Rak and then gave it back to the user (actions #13, #14, #23). This strengthens the link between the two characters.

At the same time, the narrative experience suffers from various drawbacks of the action calculus algorithms, for example,

- The course of actions lacks fluidity.
- The values of the story are not expressed in this session.
- Some actions seem quite “out of context” (#16, #18).
- Some actions lack coherence: Why Lord refuses to give the book of rhetoric to the user (#26), while he gave him/her the idea of reading this book (#21) and he previously accepted to get the hook for him/her (#14)?

- Some sequences of action are obvious, and still have to be performed by the user: deciding to get the book of rhetoric (#25) after one has decided to read it (#22).
- Some actions lack additional information: When Lord gives the hook to the user, it would be good to know how he got it, or any other hint of what happened.
- Some actions are not motivated enough (#2).

More generally, the narrative experience lacks some dramatic intensity.

Quantitative Measures of Choice

The story whose trace is represented in the Appendix contains 34 actions, including 23 actions played by the user.

At each user turn the number of choices presented to the user is measured, as well as the number of choices that are different from the previous turn. The results are displayed in Figure 5. In average, the user had 93 choices, with 7 new choices from the previous turn.

The number of choices increases in the first half of the story and then remains stable. The minimum number of choices is 37 (at the beginning of the story). Some of these choices look similar (for example, the user can ask anybody if s/he has the hook, which makes 7 different but similar actions) and their significance is variable (starting a riot is more significant than informing another character that you have got the ring, at least in this story). Nevertheless, these large amounts of choices give a freedom to the user, which is necessary for interactive drama.

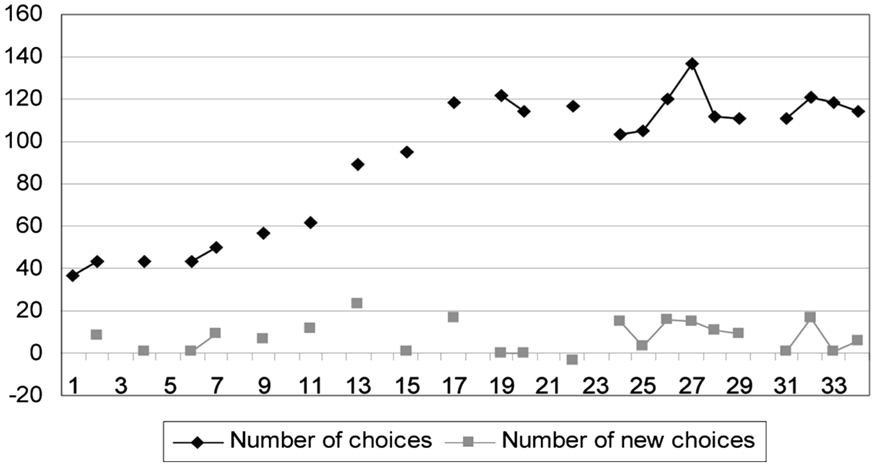

FIGURE 5 User's choices. The diamonds represent the number of choices, while the squares represent the number of new possibilities. Lines between points indicate that the user played the actions in a row, without any visible nonplayer characters intervention. Numbers in the X axis refer to the visible actions, as displayed in the Appendix.

The number of new choices varies from −3 (less choices than in the previous turn) to 23, with an average of 7. This means that not only the user proposes a large number of choices, but these choices evolve during the story. The number of new choices varies quite discontinuously in time. High values correspond to important turns in the story: Lord informs the user that s/he needs some allies (action #12, which leads to 23 new choices for action #13) or the user decides to get the book of rhetoric (action #25, which leads to 16 new choices for action #26).

DISCUSSION

The Paradox of Interactive Drama Solved?

As mentioned in the beginning of this article, the paradox of interactive drama lies in the inverse correlation between narrativity and interactivity that has been observed so far. It is also justified by the intuition that the more influence we give to the user, the less narrative control we give to the author.

Theoretically, this paradox is solved by stating that the narrative construction is not a homogeneous set of data that is to be shared between an author and a user, as a pie is shared into different pieces. The author and the user operate at qualitatively different levels. The author sets up a world of possible actions and events handled by narrative rules and algorithms s/he writes or parameterizes, while the user interacts with the mechanics created by the author. If ID were a chess game, the author would write the rules of chess, while the user would play chess. The success of interactive narrative relies on the design and implementation of these rules. Other researchers have investigated how these rules could be borrowed from pen and paper role-playing games (Louchart and Aylett 2003) and have proposed a general architecture, which has not yet been implemented (Louchart and Aylett 2004). Our architecture is based on a combination of several narrative theories. It has been implemented and tested, allowing an authentic assessment of the approach.

The implemented prototype provides a high level of interactivity. The user really influences the story. However, in terms of quality of the narrative, the experience is somewhat disappointing. Indeed, having solved the core problem of ID reveals serious issues for building a finalized version. These issues are discussed in the next three subsections.

Future Work: Interface

The current interface consists of presenting to the user all possible actions. It is rather a testing interface on the interactive narrative engine than a user-centered interface. Indeed, the interface is not usable because the user has to navigate through a huge list of choices (93 in average in the previous example). This problem of having too many choices is a counterpart of our successful attempt to give more freedom to the user.

Providing a well-designed graphical interface to navigate through the large amount of choices does not constitute a valid solution to this problem. Indeed, this would push the user into a retrieval situation, which is not suitable to the freedom ID is to provide. In a well-designed ID, the user should anticipate the set of possible actions and use the interface to build the action s/he has in mind (Szilas 2004). A taxonomy of these interfaces has been established in Szilas (2004): The two main types of interfaces are the free interfaces (free text, free speech) and the direct interfaces (menu-based). While the former suffer from the limited ability of the computer to understand humans, the latter suffers from the effort it requires from the user and the disrupting of immersion. We are currently investigating a direct interface for ID.

Future Work: Improving "Storyness"

Although the example proposed previously is a story, its narrative quality is poor. This is due in part to the limit of the model of the user, which encodes the "storyness" of the sequence of actions.

To improve it, we will follow two main approaches in parallel.

- By testing the system with real users, it is expected to get feedback of the missing features of the narrative experience. Even if various necessary improvements to the model of the user have been identified, such testing will allow to prioritize these future improvements.
- Previous work on interactive narrative and story generation provide us many suggestions for improvements. For example, the management of dramatic situations proposed in Sgouros (1999), the user modeling in terms of thought flow and activity flow (Weyhrauch 1997), or the plot units as defined in Lehnert (1981) are relevant to this work.

Future Work: Authoring

Last but not least, building a good ID requires a good story, which relies on an author entering the narrative data into the system. Unfortunately, building a well-designed authoring interface to the goal task structures depicted previously is not enough: Writing such abstract structures is fundamentally quite difficult for an author, and differs deeply from the existing writing skills (Szilas et al. 2003). This constitutes another counterpart of having solved the paradox of interactive drama.

To solve this issue, we need a methodology and tools to enable an author to navigate between the abstract representation of the story (its structures) and its concrete realization.

The following tools are currently under investigation.

- Easy-to-use interface to enter the structures and the surface forms in an integrated manner
- An integrated environment that will allow to immediately test the story data, even if they are incomplete
- Scenario validation and assessment tools, inspired from the model-checking approach

CONCLUSION: NARRATIVE AI VS. TRADITIONAL AI

In the system and approach described in this article, narrative AI (or narrative intelligence) relates to the modeling of the intelligence of a narrator. In that sense, narrative AI is just a specific application of AI.

Narrative is not limited to novels, comics, movies, games, and other entertainment artifacts: It is a fundamental way of human communication. In that sense, narrative AI is potentially a fundamental way for the computer to interact with human beings. Our findings on interactive drama are thus expected to find in the future a wider set of applications, which share the following features.

- Higher-level human-computer interaction is involved.
- Communication is considered over a certain period of time. A request in a search engine could not be an interactive narrative, but a session of several requests could.
- The interacting environment is rich and complex. A basic text editor is less likely to provide the space for an interactive narrative to occur than an operating system or a web browser that would be able to analyze the pages displayed to the user.

Another broad contribution of narrative AI is the central role that is given to the author. In traditional AI, the author is the expert, from which the system—possibly aided by a cognitive engineer—extracts some useful knowledge. In narrative AI, the focus is changed. It is the author, with the help of the system, who expresses himself towards a user. This change of focus has been discussed under the new field of "expressive AI" (Mateas 2002). We see in our own research on interactive drama the potentiality of changing the focus of several AI applications that do not belong to the field of artistic-based AI. Indeed, communication oriented subfields of AI such as embodied agents, intelligent learning environments, and natural language generation are designed in a competency-centered manner. However these fields meet some difficulties in building effectively used systems, outside of research. We conjecture that these difficulties could be alleviated if the systems were more designed towards the animators, teachers, and writers who are the authors of the specific applications.

We have not solved these authoring issues either in the field of ID. But at least the specificity of this field has opened our eyes to the necessity for applied AI to change its focus not only from the machine to the final user (a now classical claim of human-centered design), but also from the machine to the author.

Let us call this change of focus "authorable AI."

APPENDIX Transcript of a Story Played Interactively, as It Appears to the Player

1. You decide to start a riot against Nelson.
2. Jack to Malcolm: "I want to start a riot against Nelson."
3. Malcolm to Jack: "You should start a riot against Nelson."
4. Jack to Malcolm: "I have a problem to start a riot against Nelson: The gate is closed."
5. Malcolm to Jack: "The gate is closed because you haven't got the hook."
6. You decide to get the hook.
7. You ask Malcolm: "Do you have the hook?"
8. Malcolm to Jack: "I haven't got the hook."
9. You ask Rak: "Do you have the hook?"
10. Rak to Jack: "Hey man . . . I have the hook."
11. You ask for the hook from Rak. You are not a good speaker.
12. Lord to Jack: "My dear Jack, If you start to start a riot against Nelson, then be careful: The guards are armed and ready to stop anyone trying to escape . . . You need allies. . . ."
13. Jack to Lord: "I need your help! Since it is necessary to be a good speaker, I can't ask Rak for the hook. Can you do this for me?"
14. Lord to Jack: "It is a pleasure for me to get the hook for you."
15. You decide to make friends with Malcolm.
16. Lord to Jack: "My dear Jack, I have the possibility to trade the book of rhetoric with the hook from Rak."
17. You flatter Malcolm: "My dear Malcolm! You are the most valuable person here . . . Let's be friends." Malcolm answers: "Listen Jack . . . Stop your lies! If you want to be my friend, you need to deserve it. . . ."
18. Lord to Jack: "My dear Jack, I have a problem to trade the book of rhetoric with the hook from Rak: Rak might not have the hook anymore."
19. You decide to learn how to talk better.
20. Jack to Lord: "I wish to learn how to talk better."
21. Lord to Jack: "My dear Jack, you have the possibility to read the book of rhetoric."
22. You decide to read the book of rhetoric.
23. Lord gives the hook to Jack.
24. Jack to Lord: "Giving me the hook was a good choice."
25. You decide to get the book of rhetoric.
26. You ask for the book of rhetoric from Lord. You are not a good speaker.
27. You propose to Lord to exchange the ring against the book of rhetoric . . . He accepts!
28. You read the book of rhetoric. You are now a much better speaker.
29. You read the book of rhetoric. You are now a much better speaker.
30. Malcolm to Jack: "You should start a riot against Nelson."

(Continued)

APPENDIX Continued

31. You flatter Malcolm: "My dear Malcolm! You are the most valuable person here . . . Let's be friends." It seems to work, he smiles to you . . . You and Malcolm, you are friends.
32. You flatter Malcolm: "My dear Malcolm! You are the most valuable person here . . . Let's be friends." It seems to work, he smiles to you . . . You and Malcolm, you are rather close.
33. You open the gate and escape to the deck.
34. After a fierce fight with the crew, some prisonners manage to reach the cabin of Nelson and neutralize him. For his courage the new captain is chosen: "Jack is captain! Jack is captain!"

REFERENCES

Adam, J.-M. 1994. *Le Texte Narratif.* Paris: Nathan.

Aristotle, 330 BC. 1997. *The Poetics.* Mineola, NY: Dover.

Aylett, R. S. 1999. Narrative in virtual environments–towards emergent narrative. In: *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 83–86, North Falmouth, Massachusetts, USA: AAAI Press.

Bailey, P. 1999. Searching for storiness: Story-generation from a reader's perspective. In: *Proc. AAAI Fall Symposium on Narrative Intelligence* pages 157–164, North Falmouth, Massachusetts, USA: AAAI Press.

Barthes, R. 1968. Introduction à l' analyse structurale des récits. *Communications* 8:8–27.

Bates, J., B. Loyall, and S. W. Reilly. 1992. Broad agents. *Proceedings of the AAAI Spring Symposium on Integrated Intelligent Architectures* 2(4).

Bates, J. 1992. *The Nature of Character in Interactive Worlds and The Oz Project.* Pittsburgh, PA: Technical Report CMU-CS-92–200, School of Computer Science, Carnegie Mellon University.

Bremond, C. 1974. *Logique durécit.* Paris: Seuil.

Bringsjord, S. and D. Ferrucci. 1999. BRUTUS and the narrational case against church's thesis. In: *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 105–111, North Falmouth, Massachusetts, USA: AAAI Press.

Busemann, S. and H. Horacek. 1998. A flexible shallow approach to text generation. In: *Proceedings of the Nineth International Natural Language Generation Workshop (INLG' 98)*, Niagara-on-the-Lake, Ontario, Canada.

Carroll, N. 2001. *Beyond Aesthetics.* Cambridge University Press.

Cavazza, M., F. Charles, and S. J. Mead. 2001. Characters in search of an author: AI-based virtual storytelling. In: *Proceedings of the First International Conference on Virtual Storytelling (ICVS 2001). Lecture Notes in Computer Science*, 2197: 145–154.

Crawford, C. 1982. *The Art of Computer Game Design.* Reprinted online at: http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html

Crawford, C. 1996. *Is Interactivity Inimical to Storytelling?* http://www.erasmatazz.com/library/Lilan/inimical.html (accessed Sept. 2006).

Crawford, C. 1999. Assumptions underlying the Erasmatron interactive storytelling engine. In: *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 112–114, North Falmouth, Massachusetts, USA: AAAI Press.

Deemter, V. D., E. Krahmer, and M. Theune. 2001. Plan-based vs. template-based NLG: A false opposition? In: *Proceedings of the Workshop "May I Speak Freely?", in conjunction with the 23rd German Conference on Artificial Intelligence*, pages 1–5.

Donikian, S. and J.-N. Portugal. 2004. Writing interactive fiction scenarii with DraMachina. In: *Proceedings TIDSE'04, Lecture Note in Computer Science*, 3105: 101–112.

Eco, U. 1979. *Lector in Fabula.* Milan: Bompiani.

Egri, L. 1946. *The Art of Dramatic Writing.* New York: Simon & Schuster.

Elam, K. 1980. *The Semiotics of Theatre and Drama.* London and New York: Routledge.

Field, S. 1984. *Screenplay – The Foundations of Screenwriting*, 3rd edn. New York: Dell Publishing.

Geldof, S. 1999. Templates for wearables in context. In: *Proceedings of the Workshop on Natural Language Generation, German Annual Conference on AI*, pages 48–51, KI-99, Bonn, Germany.

Genette, G. 1969. *Figures II.* Paris: Seuil.

Genette, G. 1972. *Figures III.* Paris: Seuil.

Greimas, A. J. 1970. *Du Sens.* Paris: Seuil.

Jhala, A. and R. M. Young. 2005. A discourse planning approach for cinematic camera control for narratives in virtual environments. In: *Proceedings of the National Conference of the American Association for Artificial Intelligence*, pages 307–312, AAAI Press.

Jouve, V. 2001. *Poétique des Valeurs.* Paris: Presses Universitaires de France.

Juul, J. 1999. *A Clash between Computer Game and Narrative.* Master thesis, IT University of Copenhagen.

Kelso, M., T. P. Weyhrauch, and J. J. Bates. 1992. *Dramatic Presence.* TR CMU-CS-92–195, Pittsburgh, Carnegie Mellon University.

Klesen, M., J. Szatkowski, and N. Lehmann. 2000. The black sheep – interactive improvisation in a 3D virtual world. In: *Proceedings of the i3 Annual Conference*, pages 77–80.

Lang, R. 1999. A declarative model for simple narratives. In: *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 134–141, North Falmouth, Massachusetts, USA: AAAI Press.

Lavandier, Y. 1997. *La Dramaturgie.* Cergy, France: Le Clown et l'Enfant.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 4:293–331.

Louchart, S. and R. Aylett. 2002. Narrative theories and emergent interactive narrative. In: *Proceedings of Narrative and Learning Environment Conference*, (NILE 02) pages 1–8, Edinburgh, Scottland.

Louchart, S. and R. Aylett. 2003. Solving the narrative paradox in VEs – Lessons from RPGs. In: *Proceedings for the the 4th International Working Conference on Intelligent Virtual Agents (IVA'03). Lecture Notes in Computer Science*, 2792: 244–249.

Louchart, S. and R. Aylett. 2004. Emergent narrative, requirements and high-level architecture. In: *Proceedings of the 3rd Hellenic Conference on Artificial Intelligence* (SETN04) pages 298–308, Pythagorion, Greece.

Majerko, B. and J. Laird. 2003. Building an interactive drama architecture. In: *Proceedings TLDSE'03*, pages 226–237.

Mateas, M. 2002. *Guiding Interactive Drama.* Ph.D. Dissertation, Tech report CMUCS-02–206, Carnegie Mellon University.

Mateas, M. and A. Stern. 2000. Towards integrating plot and character for interactive drama. In: *Proc. AAAI Fall Symposium on Socially Intelligent Agents: The Human in the Loop*, pages 113–118, North Falmouth, Massachusetts, USA: AAAI Press.

McKee, R. 1997. *Story: Substance, Structure, Style, and the Principles of Screenwriting.* New York: HarperCollins.

Pelachaud, C. and M. Bilvi. 2003. Computational model of believable conversational agents. In: *Communication in MAS: Background, Current Trends and Future*, ed. Marc-Philippe Huget, Springer-Verlag.

Propp, V. 1928/1970. *Morphologie du Conte.* Paris: Seuil.

Reiter, E. 1995. NLG vs. templates. In: *Proceedings of the Fifth European Workshop on Natural Language Generation*, pages 95–105, Leiden, The Netherlands.

Reiter, E. and R. Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3(1):57–87.

Riedl, M., C. J. Saretto, and M. Young. 2003. Managing interaction between users and agents in a multiagent storytelling environment. In: *Proceedings of the Second International Conference on Autonomous Agents and Multi-Agent Systems.* July 14–18, 2003, Melbourne, Victoria, Australia, ACM Press, 741–748.

Russel, S. and P. Norvig. 2003. *Artificial Intelligence: A Modern Approach.* 2nd edition. Saddle River, NJ: Prentice Hall.

Ryan, M.-L. 2001. Beyond myth and metaphor – The case of narrative in digital media. *Game Studies the International Journal of Computer Game Research* 1:1.

Schneider, S. 2006. The paradox of fiction. In: *The Internet Encyclopedia of Philosophy*, http://www.iep.utm.edu/f/fict-par.htm (accessed Sept. 2006).

Sengers, P. 1998. Do the thing Right: An Architecture for action-expression. In: *Proc. Autonomous Agents '98*, Minneapolis, MN: ACM Press.

Sgouros, N. M. 1999. *Dynamic Generation, Management and Resolution of Interactive Plots, Artificial Intelligence*, 107(1):29–62.

Somanchi, S. K. 2003. *A Computational Model of Suspense in Virtual Worlds.* Technical Report Number 03-002, Liquid Narrative Group, North Carolina State University.

Spierling, U., D. Grasbon, N. Braun, and I. Iurgel. 2002. Setting the scene: Playing digital director in interactive storytelling and creation. *Computer & Graphics* 26:31–44.

Stern, A. and M. Mateas. 2003. Integrating plot, character and natural language processing in the interactive drama facade. In: *Proceedings TLDSE'03*, eds. Göbel et al.: Frauenhofer IRB Verlag.

Stone, M. and D. DeCarlo. 2003. Crafting the illusion of meaning: Template-Based specification of embodied conversational behavior. In: *Proceedings of the 16th International Conference on Computer Animation and Social Agents (CASA 2003)*, 11–16.

Szilas, N. 1999. Interactive drama on computer: Beyond linear narrative. In: *Proc. AAAI Fall Symposium on Narrative Intelligence*, North Falmouth, MA: Menlo Park: AAAI Press.

Szilas, N. 2001. A new approach to interactive drama: From intelligent characters to an intelligent virtual narrator. In: *Proc. of the Spring Symposium on Artificial Intelligence and Interactive Entertainment*, Stanford CA, 72–76, Menlo Park: AAAI Press.

Szilas, N. 2002. Structural models for interactive drama. In: *The Proceedings of the 2nd International Conference on Computational Semiotics for Games and New Media*, Augsburg, Germany, Sept. 2002.

Szilas, N. 2003. IDtension: A narrative engine for interactive drama. In: *Proceedings TIDSE'03*, eds. Göbel et al.: Frauenhofer IRB Verlag.

Szilas, N. 2004. Stepping into the interactive drama. In: *Proceedings TIDSE'04, Lecture Note in Computer Science 3105*, 101–112, Berlin: Springer Verlag.

Szilas, N. and M. Mancini. 2005. The control of agent's expressivity in interactive drama. In: *Proceedings of the International Conference on Virtual Storytelling (ICVS'05)*, Nov. 30–Dec. 2 2005, Strasbourg: Springer Verlag.

Szilas, N., O. Marty, and J.-H. Rety. 2003. Authoring highly generative interactive drama. In: *Proceedings of the Second International Conference on Virtual Storytelling – ICVS 2003*, Lecture Notes in Computer Science, n. 2897, 37–46, Toulouse, France: Springer Verlag.

Szilas, N. and J.-H. Rety. 2004. Minimal structures for stories. ACM workshop Story representation, mechanism and context. In: *conjunction with the 12th ACM International Conference on Multimedia*, October 2004, New York: ACM Press.

Tan, E. 1996. *Emotion and the Structure of Narrative Film: Film as an Emotion Machine.* Mahwah, NJ: Erlbaum.

Theune, M. 2000. *From Data to Speech–Language Generation in Context.* Ph.d. Thesis, Eidhoven University of Technology, The Netherlands.

Todorov, T. 1970. Les transformations narratives. *Poétiques* 3:322–333.

Vale, E. 1973. *The Technique of Screenplay Writing*, 3rd edition. London: Universal Library Edition.

Wilensky, R. 1983. Story grammars versus story points. *The Behavioral and Brain Sciences* 6: 579–623.

Weyhrauch, P. 1997. *Guiding Interactive Drama.* Ph.D. Dissertation, Tech report CMU-CS-97–109, Carnegie Mellon University.

Young, R. M. 1999. Notes on the use of plan structure in the creation of interactive plot. In *Proc. AAAI Fall Symposium on Narrative Intelligence*, pages 164–167, North Falmouth, MA, USA: AAAI Press.

Young, R. M. 2002. The co-operative contract in interactive entertainment. In: *Socially Intelligent Agents*, eds. Alan Bond et al., Boston, MA: Kluwer Academic Press.