Laboratório VISGRAF
Instituto de Matemática Pura e Aplicada

An Expressive Talking Head for an Interactive Storytelling System

*Paula Salgado Lucena Rodrigues, Cezar T. Pozzer,
Bruno Feijo, Angelo Ciarlim, Antonio Furtado, Luiz Velho.*

Technical Report TR-2006-01 Relatório Técnico

March - 2006 - Março

An Expressive Talking Head for an Interactive Storytelling System

Paula S. L. Rodrigues[*]
Department of Informatics, PUC-Rio

Cesar T. Pozzer[†]
Department of Informatics, PUC-Rio

Bruno Feijo[‡]
Department of Informatics, PUC-Rio

Luiz Velho[§]
Instituto de Matematica Pura e Aplicada

Angelo E. M. Ciarlini[¶]
Departamento de Informatica Aplicada - UniRio

Antonio L. Furtado[‖]
Department of Informatics, PUC-Rio

Abstract

In recent years, interactive storytelling systems have been proposed to give users an opportunity to generate and interact with stories. On the other hand, in traditional live storytelling, oral narration has always played an essential role to engage the audience. In this paper, we present a 3D interactive storytelling system where the user immersion is augmented by the presence of an expressive virtual narrator. Two integrated modules compose this system. The first module (LOGTELL) is a logic-based tool for story generation and dramatization, and the second one (ETH) is a real-time facial animation module. The LOGTELL module is able to generate an ample variety of interesting stories, and automatically dramatizes them by means of virtual 3D actors running on a graphical engine. Our virtual narrator is an expressive talking head that tells the story synchronously with the dramatization, using speech and facial expressions transmitting emotions. The narration has shown to be effective, not only to explain the stories being dramatized, but also to enhance their dramatic potential, making them more attractive. The resulting environment can be used in the development of applications involving childlike worlds, authoring tools, interactive TV, games, and distance learning.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—Plan execution, formation, and generation I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—Modal logic, Representations (procedural and rule-based), Temporal logic I.3.2 [Computer Graphics]: Graphics Systems—Standalone systems I.7.1 [Document and Text Processing]: Document and Text Editing—Spelling I.7.2 [Document and Text Processing]: Document Preparation—Multi/mixed media, Hypertext/hypermedia K.3.1 [Computer Education]: Miscellaneous—Computer literacy

Keywords: real-time expressive facial animation, storytelling, speech-and-scene synchronization, real-time interaction and rendering

1 Introduction

Traditional live storytelling is an interactive performance art form, wherein the teller adjusts the vocalization, wording, physical movements, gestures, and pace of the story to better meet the needs of the responsive audience. Storytelling in its new digital and interactive form combines participation, as occurs in computer games, with automatic story generation and narration. Different storytelling systems have been proposed and implemented with different focus and features [Cavazza et al. 2002] [Mateas 1997] [Young 2000] [Spierling et al. 2002] [Sgouros 1999] [Ciarlini et al. 2005]. Although the presence of a synthetic narrator should be a welcome enhancing to the digital storytelling experience, the existing literature has not duly explored this subject. Research works on digital actors [Thalmann and Thalmann 1995], graphical multimodal user interfaces [Corradini et al. 2005] [Cassell et al. 1999] [Massaro 2003], and facial animation [Parke and Waters 1996] do not address the question of synthetic narrators in interactive storytelling.

This paper describes the incorporation of a virtual narrator, capable of emotional expressions synchronized with speech, to the LOGTELL [Ciarlini et al. 2005] storytelling system. Speech and facial animation techniques were combined with plot generation, user interaction and 3D dramatization, in order to better communicate the story, increase dramatic potential and help user interaction.

A key point in the implementation of a storytelling system is whether it should be character-based or plot-based. In a character-based approach, such as those in [Cavazza et al. 2002], the storyline usually results from the real-time interaction, at any time, between virtual autonomous agents and the user. Although powerful in terms of interaction and variety of stories, such an extreme interference level may lead the plot to unexpected situations or miss essential predefined events. In contrast, in a plot-based approach, such as those in [Spierling et al. 2002], characters should follow rigid rules specified by a plot. By doing this, the coherence of the story might be guaranteed but the level of interaction is reduced.

LOGTELL combines both plot- and character-based features. It is based on the logic specification of a model for the genre, where possible actions and goals of the characters are described. Plot generation and 3D dramatization are integrated but separately treated. During dramatization, virtual actors perform the events in the plot without user interference. Nevertheless, the user can alternate phases of plot generation, in which intervention is possible, and dramatization. In this way, the two requirements are met: the generated stories are always coherent and we are not limited to a small

[*] e-mail: paula@inf.puc-rio.br
[†] e-mail:pozzer@inf.puc-rio.br
[‡] e-mail:bruno@inf.puc-rio.br
[§] e-mail:lvelho@impa.br
[¶] e-mail:angelo.ciarlini@uniriotec.br
[‖] e-mail:furtado@inf.puc-rio.br

group of predefined alternatives. Every intended alternative can be obtained using a combination of simulation and user interaction, provided that it is in accordance with the logics of the genre. The use of a virtual narrator in such an environment provides a very convenient way to explain the chaining of events, entailed by the conventions of the genre, and to transmit the emotion associated with each event.

In the proposed system, the virtual narrator is an expressive talking head implemented by the facial animation module ETH. This module receives markup-texts containing story fragments and produces, on the fly, a facial animation that gives voice to this input text. The speech is automatically generated using text-to-speech (TtS) mechanisms. The ETH module controls the lip synchronization and the emotional expressions. These expressions are obtained through the text markup parameters.

The environment has the capability to build, present and narrate different stories from different genres. The example shown here is based on a Swords and Dragons context, where heroes, victims and villains interact in a 3D scenario occupied by castles and churches. The narrator's primary duty is to tell, from a third person perspective and with the appropriate emotion, each scene of the story. The environment has the flexibility to allow the narrator to assume the role of any character in the story (in first-person discourse), and it has also the option to introduce more than one narrator with a different physical appearance (man, woman, or child) for each character.

One of the main contributions of the project discussed in the present article is the development of this environment for generating and visualizing stories involving virtual actors, as seen from the standpoint of an emotional narrator. The benefits of including one such animated emotive narrator in the interactive storytelling system seem clear. It should look particularly pleasant to a children's public, to which the presence of an expressive character is an important incentive in teaching and entertainment activities. Other fields of useful application are game development, interactive TV, and distance learning.

Section 2 presents LOGTELL's architecture and its mechanisms for building plots with user interaction. Section 3 briefly describes the ETH module, giving an overview of the facial animation tool. Section 4 initially points out aspects that could be improved in the story generation and dramatization through the use of a talking head narrator; next, it gives details of the communication between the systems, including synchronization and data exchange schemes. Section 5 discusses related work, and section 6 contains the conclusions.

2 StoryTelling Module: The Interactive LOGTELL System

LOGTELL is based on modelling and simulation. The idea behind LOGTELL is to capture the logics of a genre through a temporal logic model and then verify what kind of stories can be generated by simulation combined with user intervention. In this way, we focus not simply on different ways of telling stories but on the dynamic creation of plots. The model is composed of typical events and goal-inference rules. Plots are generated by multiple cycles of goal-inference, planning and user intervention.

Instead of creating an immersive experience in which the user takes part in the story as one of the characters, we try to explore the possibilities of generating a large variety of coherent stories. For this reason, our stories are told with a third-person viewpoint. User intervention is always indirect. During the simulation, the user can intervene either passively (weak intervention), just letting the partially-generated plots that seem interesting to be continued, or, in a more active way (strong intervention), trying to force the occurrence of events and situations. These are rejected by the system whenever there is no way to change the story to accommodate the intervention. Plot dramatization can be activated for exhibiting the final as also the partial plots. During the dramatization, characters are represented by actors in a 3D-world. During the performance of an event, low-level planning is used to detail the tasks involved in each event. In order to integrate dramatization and plot generation, we decided to implement our own graphical engine, so that we could guarantee the compatibility between the logical model of our plots and the corresponding graphical dramatization.

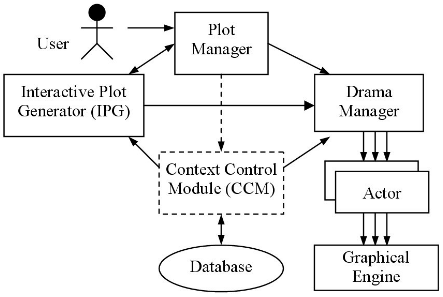

User
Plot Manager
Interactive Plot Generator (IPG)
Drama Manager
Context Control Module (CCM)
Actor
Database
Graphical Engine

Figure 1: LOGTELL's architecture.

LOGTELL comprises a number of distinct modules to provide support for generation, interaction (management) and 3D dramatization of interactive plots, as shown in Figure 1. The arrows represent the dataflow.

The Interactive Plot Generator (IPG), implemented in Prolog, performs simulations using the context specified by the user. The context contains: (a) the possible types of events, specified by means of operations with pre- and post-conditions; (b) a set of goal-inference rules, written with a temporal modal logic to specify the goals the characters might pursue when certain situatios are verified during the plot; and (c) the initial configuration of the story, describing characters and their initial state. The generation of a plot starts by inferring goals of characters from the initial configuration. Given this initial input, the system uses a non-linear planner that inserts events in the plot in order to allow the characters to try to fulfill their goals. When the planner detects that all goals have been either achieved or abandoned, the first stage of the process is finished. The partial plot then generated is presented to the user by means of the Plot Manager and can optionally be dramatized. If the user does not like the partial plot, IPG can be asked to generate another alternative. If the user accepts the plot generated so far, the process continues by inferring new goals from the situations generated in the first stage. If new goals are inferred, the planner is activated again to fulfill them. The process alternates goal-inference, planning and user interference until the moment the user decides to stop or no new goal is inferred.

The Plot Manager comprises the user graphical interface (in Java), whereby the user can participate in the choice of the events that will figure in the plot and decide on their final sequence (Figure 2). The choice of alternative compositions of events and their total order correspond to the weak kind of user intervention that can occur. Notice that IPG generates only partial orders of events, established by the chaining of pre- and post-conditions, but the 3D dramatization requires a total order, because it is relevant to the way the story is told. Stronger inteventions are also possible. The plot manager has commands to force the insertion of events and situations, seen

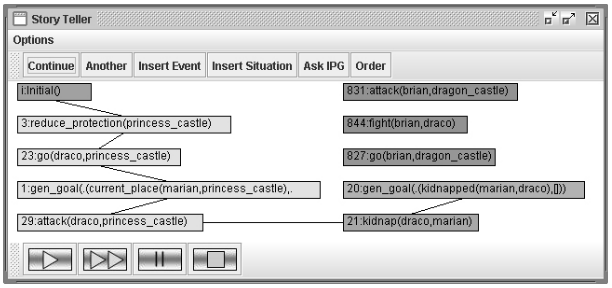

Story Teller
Options
Continue
Another
Insert Event
Insert Situation
Ask IPG
Order
i:Initial()
3:reduce_protection(princess_castle)
23:go(draco,princess_castle)
1:gen_goal(.(current_place(marian,princess_castle),.
29:attack(draco,princess_castle)
831:attack(brian,dragon_castle)
844:fight(brian,draco)
827:go(brian,dragon_castle)
20:gen_goal(.(kidnapped(marian,draco),[]))
21:kidnap(draco,marian)

Figure 2: Interface of the Plot Manager.

by IPG as additional goals. Such strong interactions are however subject to validation by IPG that tries to conciliate user interventions and the logic requirements of the genre. At any intervention phase, the user can: (a) order IPG to continue the generation process; (b) query IPG to obtain details about the situation of the story, such as the state of specific characters when a certain event occurs; or (c) order the Drama Manager to dramatize the events inserted so far.

The Drama Manager is responsible for the dramatization of the plot. It controls actors for each character in a 3D environment running on our game engine. During the dramatization, the Drama Manager consults IPG to keep the coherence between logical and graphical representations. The Drama Manager translates symbolic events into fully realized 3D visual graphical animations, guaranteeing the synchronism and logical coherence between the intended world and its graphical representation. As received from the plot generation, the plot is organized as a sequence of events, each one associated with a discrete time instant, and their effects are supposed to occur instantaneously. For the purposes of visualization, a different concept of time is used. The simulation occurs in continuous real-time and the duration of an event rendering is not previously known. Variable attributes change as the event is dramatized. In order to make logical and graphical representations compatible, the values of the variables before the dramatization of each event must agree with the pre-conditions of the event and the values at the end with its post-conditions. Actors have a geometric structure amenable to graphical representation, and a minimum of planning capabilities, at a low level of detail. During graphical representation, all control of actions each actor is supposed to perform is made by the Drama Manager. It acts as a director that coordinates sequences of linear or parallel actions performed by the whole cast. It continuously monitors the representation process, activating new tasks whenever the previous ones have been finished. As a director, it also controls the positioning of the (virtual) camera, which an option of LOGTELL permits to be transferred to the user.

3 Facial Animation Module: The Emotional ETH System

The Expressive Talking Heads (ETH) is a facial animation system which, upon receiving an input text with some special markups, is able to generate a real-time character facial animation speaking this text. ETH was developed to provide a framework for applications wherein a talking head unit may be desirable. To facilitate ETH usage, its services are available through a single façade that hides the internal organization.

Some applications have already been developed through the ETH framework, such as a 3D chat allowing the sending user to type text messages, to be listened by the receiving user from the facial animation system. Another important application is the integration with a hypermedia presentation system [Rodrigues et al. 2004]. In this application, virtual characters can be associated with other media objects, such as slide presentations, videos, subtitles, and audio tracks.

ETH has been developed using the Java Programming Language and its architecture is composed of three major modules: Input Synthesis, Face Management and Synchronization. Figure 3 presents an overview of the ETH' modular structure.

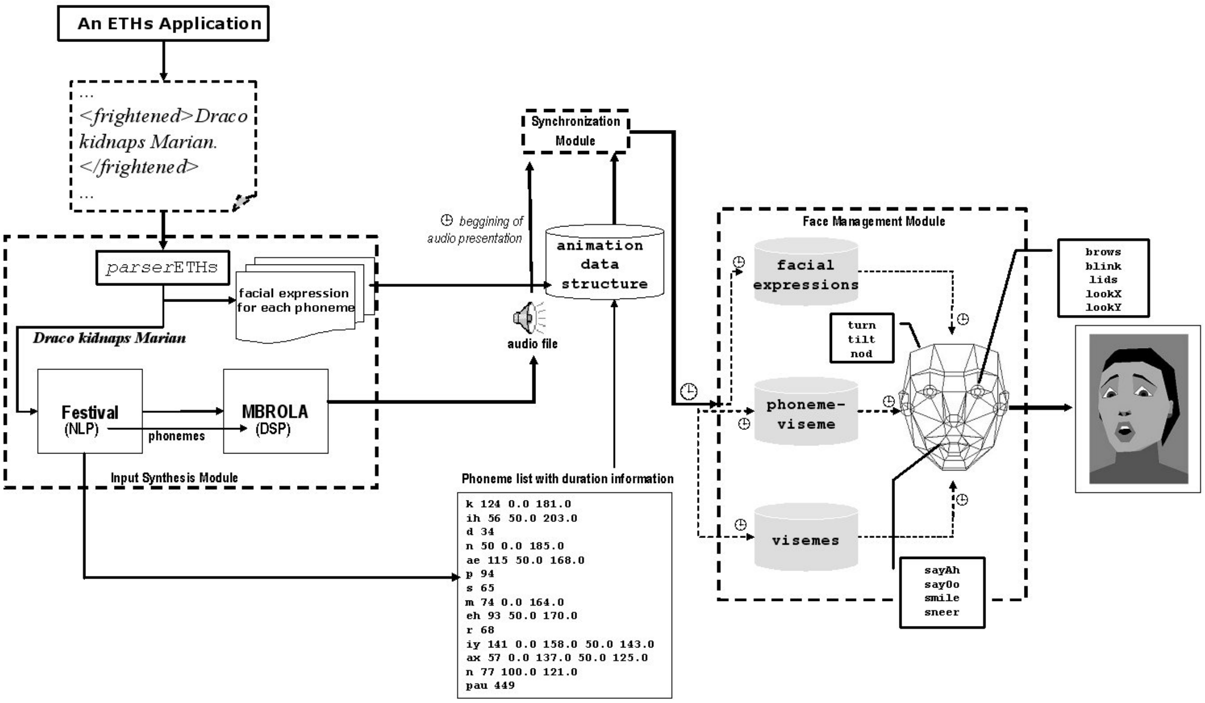
The *Input Synthesis module* is responsible for (a) capturing and treating the input text, which can have some markup elements conveying information about the character's emotion, the voice gender (currently, masculine and feminine adults), and the speech language (currently, American English and British English), and (b) for generating as output a data structure containing the fundamental units (phonemes, duration, emotion, etc.) needed to build the facial animation corresponding to the input text. The module has two submodules: a parser, responsible for separating the speech content itself (text without markups) from the speech and animation markups, and the TtS (*Text to Speech*) submodule, which builds the facial animation and lipsync data structures.

The parser interacts with the TtS (*Text to Speech*) submodule to build the facial animation and lipsync data structures. It progressively sends to the TtS submodule each fragment of the input markup text. The TtS submodule in ETH is made up of two independent subsystems: Festival [Black and Taylor 2004] and MBROLA [Dutoit and et al. 1998] as shown in Figure 3. In this blend of two synthesizers, Festival works as the Natural Language Processing unit (NLP), being responsible for the speech phonetic description creation (list of phoneme entries, each one containing the phoneme label, duration and pitch), while the MBROLA works as the Digital Signal unit (DSP), in charge of generating the final output audio file. The advantage of putting them together is the acquisition of a TtS synthesizer that offers a multilingual platform (MBROLA's contribution). This flexibility is important to Expressive Talking Heads, because it allows the user to select language and gender as system parameters, thereby enhancing the system expressiveness.

Working on the phonetic structure, the parser can identify the phonemes corresponding to each marked text fragment, and can then assign, for example, the set of phonemes for each emotion in the speech. All these results are stored in the animation data structure, which will be queried by the synchronization module.

The second ETH module, the *Face Management module*, is connected to another external subsystem, named Responsive Face [Perlin 1997]. The ETH face was inherited from this subsystem, which defines a three-dimensional polygonal mesh. The Face Managemente module controls the Responsive Face subsystem by means of a set of minimal controls, but this minimal set is complete enough to provide a fair degree of expressivenes to the talking head.

The face is animated by the application of relax and contract commands over the mesh edges (face muscles). ETH improves the Responsive Face features, adding the concept of visemes. Viseme is the name given to a mouth configuration for a specific phoneme. When initializing the system, the face management module builds a table of 16 visemes and 8 facial expressions. Each table entry stores the values for contracting/relaxing the face corresponding muscles commanding the Responsive Face. The face management also builds a table defining the phoneme-viseme mapping. During the animation process, the module receives requests from the Synchronization module (see below) to supply information from such



```
k 124 0.0 181.0
ih 56 50.0 203.0
d 34
n 50 0.0 185.0
ae 115 50.0 168.0
p 94
s 65
m 74 0.0 164.0
eh 93 50.0 170.0
r 68
iy 141 0.0 158.0 50.0 143.0
ax 57 0.0 137.0 50.0 125.0
n 77 100.0 121.0
pau 449
```

Figure 3: An overview of the ETH architecture and modules.

databases to activate facial muscles in order to produce the facial expressions desired.

The third and last ETH module is the *Synchronization module*, which is responsible for the fine synchronization between speech and facial muscle movements. Parallel to the audio file reproduction, the synchronizer polls the audio controller to check the effective playing instant. Using the information in the animation data structure, the Synchronizer discovers the current phoneme and the current character emotion. Then it asks the Face Manager to animate the face in order to achieve the corresponding viseme and facial emotion. The Face Manager gets the associated viseme and facial expression muscle contracting/relaxing values, and asks the Responsive Face to apply these values on the mesh. In reality, instead of working just with the current phoneme, the Synchronizer uses diphones (two consecutive phonemes interpolated by their durations), since the lip positions for the same phoneme generally change according to the phoneme context (speech co-articulation aspect). The Synchronization module also includes components to control the movements of the head and of the eyes, so as to produce a more natural output.

4 The Virtual Narrator in Action

By integrating ETH and LOGTELL, the plot generation and dramatization processes became more intuitive and accessible. The new interface is expected to look friendlier to users guiding the composition of a story plot, or merely attending as spectators, and even to authors working on the specification and revision of the story genre in use. Figure 4 illustrates the communication between LOGTELL and ETH using the narrator.

The ETH system, by means of live audio synchronized with a 3D emotive virtual narrator, provides an additional medium to communicate information. During plot generation, it can be used to complement what is presented, perhaps too concisely, in dialog text boxes. During dramatization, the virtual narrator can be used not only to read aloud the subtitles narrating the current action, but also to explain what is happening and reveal what lies “behind the scene”. This is possible because all metadata, i.e. the internal definition of the genre, especially the pre-conditions and post-conditions of operations and the goal-inference rules, stay available at runtime. In particular, it should be pointed out that the ability of expressing emotions during dramatization is essential to increase the dramatic potential of the story. The result is a more engaging experience with a better comprehension of the story by the spectators.

The complementary explanation provided by ETH can be either produced in real-time, or pre-synthesized and later inserted in the appropriate context. The real-time strategy favours the necessary flexibility to help the user during plot generation. The user could, for instance, query IPG about details of a specific character at a certain point of the story. The answer to be given by the narrator would also have to be generated at runtime. On the other hand, when Dramatization is activated, since (part of) the story to be told does not change, parallelism can be used to pre-synthesized speech for the next events while a previous one is being presented. In this way, CPU processing time can be saved and more attention can be paid to information contents, communicative efficacy and stylistic quality.

4.1 Graphical and Narrator Output

The graphical engine supports real-time rendering of the 3D elements. The rendering of these elements is controlled by the Drama Manager. Characters in a generated plot are regarded as actors for the dramatization. The Drama Manager acts then as the director without having to perform any intelligent processing with respect to plot generation. It essentially follows the ordered sequence of

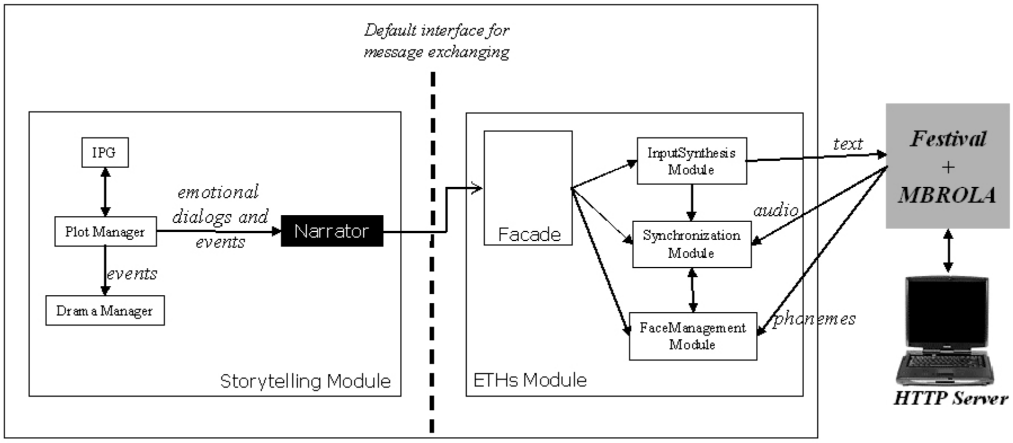


Figure 4: Overview of the whole environment developed to generate and rendering interactive stories using a narrator.

events generated at preceding stages of simulation and interaction. Each actor is implemented as a materialized reactive 3D agent, with minimal planning capabilities necessary to play their role within an event. The Drama Manager controls, from a third-person perspective, the scene and the current actors' aspect and movements. Steering behaviours [Reynolds 1999] are used by the actors for real-time interactions with the scene and, occasionally, with other actors.

When accompanying dramatization, the virtual narrator is responsible for synchronously narrating the ongoing actions being performed by the actors. In our test scenario, we use a small sub-class of the popular Swords and Dragons genre. The participants are a Princess, called Marian (the potential victim), Draco the dragon as a villain, and two heroes, the knights Brian and Hoel. Currently, we make use of simple templates (Prolog lists intercalating variables and fixed character strings) to translate the formal terms denoting the events into the natural language sentences that are used for narration. In our example, these are some of the subtitles automatically generated by the system, to be recited by the virtual narrator while the respective scene is being visualized:

1. The protection of the Princess's castle is reduced.
2. Draco kidnaps Marian.
3. Brian kills Draco.
4. Brian frees Marian.
5. Brian and Marian get married.

Since the rendering duration of most of the actions can be previously ascertained, and is usually not less than 10 seconds, the virtual narrator has enough time to describe the events being dramatized, and to add relevant contextual information. This extra material can be readily extracted by IPG, which has access to properties of characters and places at each state reached by the plot simulation, as also to the logical specification of the genre (which stays available online, as said before). The logical chaining of events, determined by specified causes, effects and goals, is an essential part of the narration. As an example, consider the event that portrays the abduction of the victim (Princess Marian) by the villain (Draco). A pre-condition for this event is the fragility of the victim and, as post-condition, the kidnapped princess is confined to the villain's castle. What ultimately motivates the event is the villain's goal of kidnapping unprotected victims. Since one of the heroes' goals is to free damsels in distress, usually in order to marry them as a reward, the kidnapped victim's situation arouses in the hero the desire (goal) of rescuing her. The simulated execution of a plan to achieve this goal leads, in turn, to a new state wherein other goals are inferred, thus causing the story to move forward. We have already implemented a text generation module that generates this kind of explanation [Furtado and Ciarlini 2000]. We are however working on text stylistic improvements to better incorporate the text generation feature to the environment.

Table 1: Operation-and-Emotion Mapping.

Operation	**Emotion**
`go(CH, PL)`	natural
`reduceprotection(VIC,PL)`	annoyed
`kidnap(VIL,VIC)`	frightned
`attack(CH,PL)`	surprised
`fight(CH1,CH2)`	angry
`kill(CH1,CH2)`	angry
`free(HERO,VIC)`	happy
`marry(CH1,CH2)`	happy

ETH is responsible for dialog synthesis, in real time, and also for handing over the speech audio and the phoneme sequences to be spoken in a synchronized way. For each phoneme, there is an associated viseme, and to visualize the viseme the narrator facial muscles are moved, as mentioned in Section 3. Each user's operation to build the story is hitched with an emotion. This emotional information is used by the virtual narrator in the exact instant when it tells the story. On the other hand, it knows that, for each word, sentence or paragraph, there is a facial expression. Internally, operations must somehow be mapped into emotions, for example as indicated in Table 1 [1].

Another important enhancement for telling the story could be introduced by describing effects of the actions in a more dramatic way, conveying the appropriate emotion. The event "Brian frees Marian" has a side-effect that is essential for understanding the story: the level of affection of the princess for the hero is raised to 100. For instance, it would be easy to apply a simple conditional template, in Prolog notation, such as:

Consulting such a template, a sentence, with appropriate emotion tags, could be sent to the virtual narrator, inducing it to comment, with a happy smile: Marian feels now a perfect love for Brian.

[1] In Table 1, CH is a character, PL is a place, VIC is a victim, VIL is a villain, and HERO is a hero.

```
template(affection(CH1,CH2,Level),
   [CH1,’ feels now ’, Aff, ’ for ’,CH2]) :-
      (Level = 0, !, Aff = ’absolutely nothing’;
      Level =< 50, !, Aff = ’a moderate liking’;
      Level =< 99, !, Aff = ’some tenderness’;
      Level = 100, Aff = ’a perfect love’).
```

In addition to speech, it is also possible to incorporate a background music line. The music can play throughout different narration phases, reflecting the varying emotions associated with the events.

Figure 5 illustrates the visual aspect of the environment.

4.2 Implementation Issues

When a user requests plot dramatization, each event is processed separately from the others, according to the connected sequence drawn by the user in the Plot Manager interface. The dramatization process involves the delivery of all specific data associated with the current event to both the Drama Manager and the narrator. In order to do that, for each individual event, the Drama Manager initially consults the IPG module to obtain the information required for describing the event, including subtitles and dialogs. The Drama Manager determines when an event dramatization has been finalized, and, in this case, requests a new one from the Plot Manager. All modules are implemented in Java, except the Drama Manager, which is implemented in C++/OpenGL.

The Drama Manager receives the event itself, and its corresponding subtitle. Its job is to transform the event into graphical animation. It coordinates the virtual actors in order to ensure accurate representation. The narrator receives the event and dialogs describing the scene. One must recall that, before starting the telling activity, it is necessary to create the corresponding speech, sending the text to the synthesizer. The synthesized text may be a simple subtitle, a text with emotion, or a concatenation of such texts corresponding to a sequence of events (the associated emotions being obtained by consulting the operation-emotion mapping table).

The Drama Manager and the narrator must be initialized before plot dramatization. The initialization of the Drama Manager comprises the tasks of constructing the scenario and loading the virtual actors. These remain in a standing state until some specific action is delegated. The initialization of the narrator includes the initialization of the façade, which manages the other main modules of ETH (Input Synthesis, Face Management and Synchronization modules). During the narrator initialization it is possible to define certain features of the virtual agent, such as the server synthesized machine, the face gender (masculine or feminine), the output voice quality and the initial facial expression.

The user may select whether he would like to see the story with 3D scenes and a narration, or only with 3D scenes. The default option is the first one, with both visual and speech narrations. With the purely visual option, the narrator is simply not created.

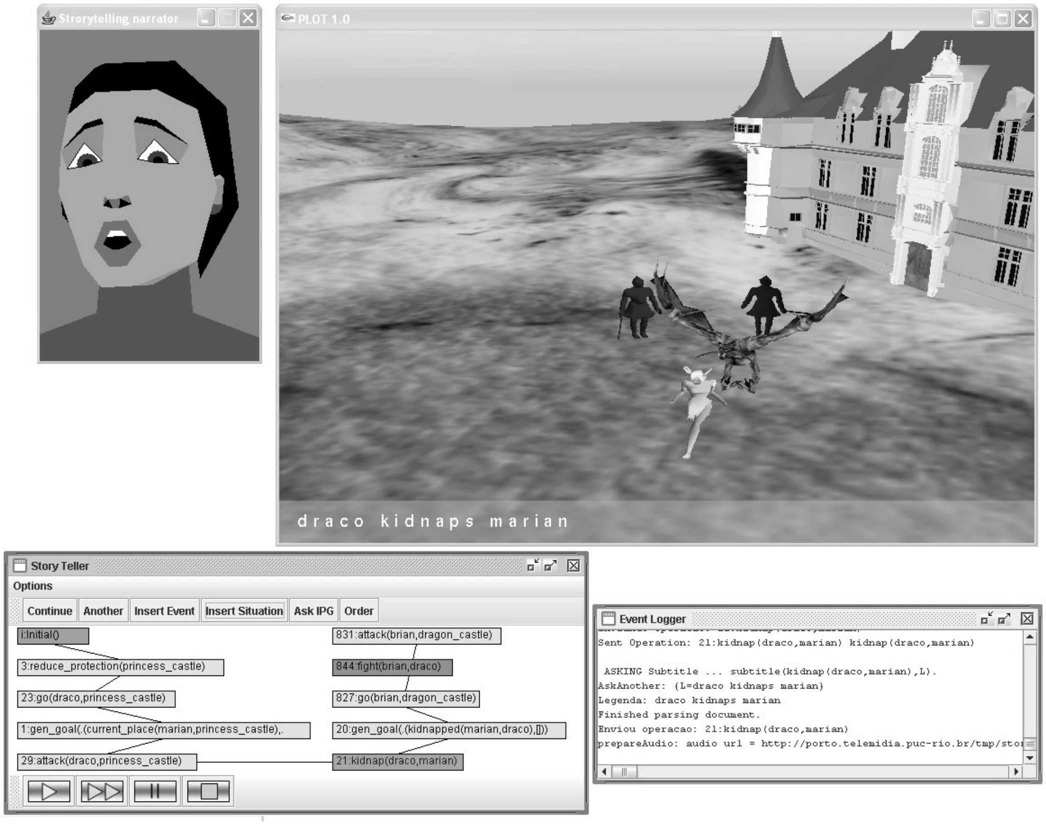
5 Related Work

The work [Silva et al. 2001] focuses on automatic plot creation, but without any kind of user interaction. The referenced paper describes a virtual storyteller framework, where plots are not predefined but created by the actions of the characters, guided by a virtual director. The virtual director is a separate agent who has general knowledge about plot structure. Both the characters (or ’actors’) and the director are implemented as intelligent agents, capable of reasoning within their own domain of knowledge. The characters can make plans to achieve their personal goals using story-world knowledge, i.e. knowledge about their virtual environment and the actions they can take in it. The director is able to judge whether the intended action of a character fits into the plot structure, using both story world knowledge and general knowledge about what constitutes a ’good’ plot. However, the virtual director is not endowed with any kind of speech, it uses text balloons to present the narrative. Another limitation is the absence of emotional facial expressions, since the narrator always presents the same behavior and attitude.

In [Theune et al. 2003] a storyteller is described, which is a synthetic character, immersed in a 3D virtual world. The aim of this character is to narrate the content of a story in a natural way, expressing the proper emotional state as the story progresses. The storyteller simply acts as a virtual narrator who reads an input text enriched with control tags. Such tags allow the storywriter to control the emotional state and behavior of the character, and the surrounding environment (set and illumination). The idea is to move the storyteller throughout the different places mentioned in the story, establishing thereby a correlation between the story and the ambience. It should be noted that the environment where the virtual narrator can be is very limited: he can stay inside a house or in a city street. The narrator is explicitly and externally controlled by the input file, which works almost as a scripting language. Nevertheless, the virtual character can assume only simple expressions, without any kind of synchronization treatment between the speech and facial movements. As a consequence, from time to time the absence of lip synchronization is very noticeable.

The work of Spierling [Spierling et al. 2002] adopts a modular system approach for interactive storytelling. In that work, a prototype of a virtual character was developed, which stays in a digital newsstand, interacting with the user through dialogs. The system appears to have originated from a fairy-tales background, but its context was adapted for business dialog, giving information about products. The scope of the work looks somewhat restricted when compared with the environment proposed in the present article. It does not offer a dramatization module for 3D scenes, the avatar acting is tailored to the business world and product selling, and, more importantly, the user interaction is limited and coordinated by the narrator. With that, the user has no freedom to modify the story direction. On the other hand, its modular design opens the possibility to integrate a graphical view module, which would allow the system to become more complete and better adapted to the proposed environment.

Finally, there is a group of interesting facial animation systems that are not associated with any kind of storytelling system [Zhang et al. 2003] [Bui et al. 2004] [Pandzic and Forchheimer 2002]. These works are correctly defined as facial animation tools, but some of them, like [Zhang et al. 2003], provide no treatment of speech, the character being limited to display emotional expressions. Some avatars based on the MPEG-4 standard, described in [Pandzic and Forchheimer 2002], have potential to be used as virtual narrators in storytelling systems. Unfortunately the MPEG-4 facial animation framework suffers from a limitation: the system loses portability and platform independence, because the framework requires an encoder and a decoder to propagate (send and receive) MPEG-4 facial animation streams. Reference [Bui et al. 2004] [Bui et al. 2003] presents a 3D talking head system with speech synchronization. The work is still under development, but it seems to have the necessary infrastructure to become a virtual narrator for a storytelling system, with features similar to those presented in this paper in connection with ETH.


Strorytelling narrator
PLOT 1.0
draco kidnaps marian
Story Teller
Options
Continue
Another
Insert Event
Insert Situation
Ask IPG
Order
i:Initial()
3:reduce_protection(princess_castle)
23:go(draco,princess_castle)
1:gen_goal(.(current_place(marian,princess_castle),.
29:attack(draco,princess_castle)
831:attack(brian,dragon_castle)
844:fight(brian,draco)
827:go(brian,dragon_castle)
20:gen_goal(.(kidnapped(marian,draco),[]))
21:kidnap(draco,marian)
Event Logger

```
Sent Operation: 21:kidnap(draco,marian) kidnap(draco,marian)

 ASKING Subtitle ... subtitle(kidnap(draco,marian),L).
AskAnother: {L=draco kidnaps marian}
Legenda: draco kidnaps marian
Finished parsing document.
Enviou operacao: 21:kidnap(draco,marian)
prepareAudio: audio url = http://porto.telemidia.puc-rio.br/tmp/sto
```

Figure 5: A snapshot of the environment.

6 Conclusion

In this paper, we present a 3D interactive environment for story generation and dramatization, using an expressive avatar to augment user immersion and emotional experience. Besides its usage in entertainment, the developed system can be adapted and applied in areas such as authoring stories, training, news presentation, distance learning and e-commerce. The system flexibility for incorporating different kinds of modules increases its ability to cope with an ample variety of applications.

The environment presented in this article has two main components: a plot-based storytelling system, called LOGTELL, and an expressive talking head system, called ETH. In the environment, the facial avatar is used as a story narrator, integrated with a 3D rendering module, with voice output generated on the fly. Moreover, the avatar exhibits emotional facial expressions in order to enhance the user perception during storytelling.

A first enhancement to the proposed system can be the modification of the text generation module to include stylistic improvements combined with automatic generation of emotion tags. Other enhancement will be the investigation about how our narrator capabilities can be fully used to co-operate with the user during plot generation. Besides working on these enhancements, we intend explore the idea of expressive avatars in other ways in the same environment. For example, they might be assigned roles as characters of the stories. Multiple instances could be generated, so as to have a number of expressive avatars interacting with each other, with an automatically generated playing performance, which is one research line under development. Another future work is the use of emotional and reacting agents to model character behaviors. An intriguing possibility is to have the avatars (working as narrators or characters) interacting vocally with user audiences, allowing users to conduct the story through their intermediacy. As an even more ambitious objective for future work, we intend to investigate how this kind of machinery could be upgraded to the point of providing useful help in the story authoring task.

Finally, we also intend to integrate our storytelling environment with an authoring and presentation hypermedia system. The goal is to model the generated stories as hypermedia documents, with the choice points in the plot represented as hyperlinks. This correspondence should permit to take advantage of hypermedia conceptual models to formally describe the synchronization among various story resources (e.g. sound tracks, flashback videos, animations, and subtitles) [Rodrigues et al. 2004].

7 Acknowledgements

The authors would like to thank CNPq and Finep for financial support under the form of PhD scholarships and research grants, in special through the contracts CNPq PQ Proc. 305982/2003-6, SEPIN-CNPQ-FINEP No. 01.02.0235.00 (Ref. 2425/02) and FINEP No. 01.04.0945.00 (Ref. 3110/04). Thanks are also due to Ken Perlin for his valuable collaboration on “Responsive Face”.

References

BLACK, A., AND TAYLOR, P., 2004. Festival speech synthesis. Software Package, version 2.0.

BUI, T., HEYLEN, D., AND NIJHOLT, A. 2003. Improvements on a simple muscle-based 3d face for realistic facial expressions. In *Proceedings 16th International Conference on Computer Animation and Social Agents (CASA'2003)*, IEEE Computer Society, 33–40.

BUI, T. D., HEYLEN, D., AND NIJHOLT, A. 2004. Combination of facial movements on a 3d talking head. In *Computer Graphics International*, 284–291.

CASSELL, J., BICKMORE, T. W., BILLINGHURST, M., CAMPBELL, L., CHANG, K., VILHJALMSSON, H. H., AND YAN, H. 1999. Embodiment in conversational interfaces: Rea. In *Proceedings of CHI*, 520–527.

CAVAZZA, M., CHARLES, F., AND MEAD, S. 2002. Character-based interactive storing. *IEEE Inteligent Systems, special issue on AI in Interactive Entertainment 17(4)*, 17–24.

CIARLINI, A., AND FURTADO, A. 2002. Understanding and simulating narratives in the context of information systems. In *Proceedings of the 21st International Conference on Conceptual Modeling - ER'2002*, 291–306.

CIARLINI, A., POZZER, C. T., FURTADO, A., AND FEIJÓ, B. 2005. A logic-based tool for interactive generation and dramatization of stories. In *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology - ACE 2005*.

CORRADINI, A., MEHTA, M., BERNSEN, N., AND CHARFUELAN, M. 2005. Animating an interactive conversational character for an educational game system. In *Proceedings of the 10th Int. Conf. on Intelligent User Interfaces*, 183–190.

DUTOIT, T., AND ET AL., 1998. The mbrola project. Software Package. URL: *http://tcts.fpms.ac.be/synthesis/introtts.html* (last accessed at September, 16, 2005).

FURTADO, A. L., AND CIARLINI, A. E. M. 2000. Generating narratives from plots using schema information. In *NLDB'00 Applications of Natural Language to Information Systems*, Springer-Verlag, London, UK, 17–29.

MASSARO, D. W. 2003. A computer-animated tutor for spoken and written language learning. In *Proceedings of the 5th Int. Conf. on Multimodal Interfaces, ICMI 2003*, 172–175.

MATEAS, M. 1997. An oz-centric review of interactive drama and believable agents. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

PANDZIC, I., AND FORCHHEIMER, R. 2002. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. ISBN 0-470-84465-5.

PARKE, F. I., AND WATERS, K. 1996. *Computer Facial Animation*, 1st ed. AK Peters Ltd.

PERLIN, K. 1997. Responsive face. Tech. rep., Media Research Lab, New York University, USA. URL: *http://mrl.nyu.edu/˜perlin/demox/Face.html* (last accessed at September, 15,2005).

REYNOLDS, C., 1999. Steering behaviors for autonomous characters.

RODRIGUES, R., LUCENA-RODRIGUES, P., FEIJÓ, B., VELHO, L., AND SOARES, L. 2004. Cross-media and elastic time adaptive presentations: the integration of a talking head tool into a hypermedia formatter. In *Adaptive Hypermedia and Adaptive Web-Based Systems, 2004, Eindhoven. Lecture Notes in Computer Science (LNCS 3137)*, 215–234.

SGOUROS, N. 1999. Dynamic generation, managing and resolution of interactive plots. *Artificial Intelligence 107*, 29–62.

SILVA, A., VALA, M., AND PAIVA, A. 2001. The storyteller: Building a synthetic character that tells stories. In *Workshop on Representing, Annotating, and Evaluating Non-Verbal and Verbal Communication Acts to Achive Contextual Embodied Agents, at Autonomous Agents Conference*.

SPIERLING, U., BRAUN, N., IURGEL, I., AND GRASBON, D. 2002. Setting the scene: playing digital director in interactive storytelling and creation. *Computer and Graphics 26*, 31–44.

THALMANN, N. M., AND THALMANN, D. 1995. Digital actors for interactive television. In *Proceedings of the IEEE (Special Issue on Digital Television, Part 2)*, vol. 83, 1022–1031.

THEUNE, M., FAAS, S., NIJHOLT, A., AND HEYLEN, D. 2003. The virtual storyteller: Story creation by intelligent agents. In *Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, 204–215.

YOUNG, R. 2000. Creating interactive narrative structures: The potential for ai approaches. In *AAAI Spring Symposium in Artificial Intelligence and Interactive Entertainment*, AAAI Press, Palo Alto, California.

ZHANG, Q., LIU, Z., GUO, B., AND SHUM, H. 2003. Geometry-driven photorealistc facial expression synthesis. In *2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 177–189.