A Model for Interactive TV Storytelling

Marcelo M. Camanho[1] Angelo E. M. Ciarlini[1]
Antonio L. Furtado[2] Cesar T. Pozzer[3] Bruno Feijó[2]

[1]UNIRIO, Dep. Informática Aplicada, Brazil
[2]PUC-Rio, Dep. Informática, Brazil [3]UFSM, Dep. Eletrônica e Computação, Brazil

Abstract

Interactive storytelling systems are applications to generate and dramatize interactive stories. The main challenge to such systems is the generation of coherent stories while users are watching and interfering with what is happening. In an interactive TV environment, quality and diversity of narratives are crucially important objectives. In addition, new requirements related to comfort in user interaction, responsiveness and scalability have to be taken into account. In this paper, we present a model for interactive TV storytelling to cope with these requirements. The model was implemented in a new version of the planning-based interactive storytelling system Logtell.

Keywords: Interactive TV, Interactive Storytelling, Modeling and Simulation, Planning, Multimedia

Authors' contact:
marcelo.camanho@uniriotec.br
angelo.ciarlini@uniriotec.br
furtado@inf.puc-rio.br
pozzer@inf.ufsm.br
bfeijo@inf.puc-rio.br

1. Introduction

Interactive storytelling systems are computer applications for telling stories that can be modified to some extent by their users. While in conventional games stories are essentially used to create challenges for the player, in interactive storytelling applications stories are expected to surprise and entertain. As a consequence, the quality of the stories in terms of coherence and dramatic content must be regarded as a prime concern. Different approaches for interactive storytelling have been proposed and implemented with different goals. Some of them are more directed to games and others to filmmaking and literature. One of the main challenges for the implementation of such systems continues to be the conciliation of a good level of interactivity with the coherence of the stories.

In recent years, traditional Analog TV is gradually being replaced by Digital TV, with higher sound and image quality, and with new interaction possibilities. Besides that, we have seen an ample dissemination of new communication media, such as broadband Internet and 3G mobile phones, which offer competing alternatives for the exhibition of shows that would, in principle, seem naturally fit for TV.

The phrase interactive TV (iTV) has been used both in the context of open Digital TV and other media featuring some kind of interactivity for TV programs. As TV is one of the classical media for telling stories, new possibilities of interaction open great opportunities for the creation of interesting applications. In this context, however, certain requirements tend to become even more essential, such as quality, coherence and diversity of stories, as well as the need for comfortable and simple interaction methods. It is necessary to find hybrid means for presenting stories on iTV, mixing features of games and conventional TV. On the one hand, the appeal of TV programs for regular spectators has to be maintained, but, on the other hand, various ways of interacting with the medium should be provided. And one must consider not only the case in which users want to actively intervene in a story, but also the case in which they just want to watch TV without being called to interact by any means.

When spectators watch a film on TV, their satisfaction is directly related to the quality of the story, and coherence is a crucial issue for a good story. Interaction methods should not violate coherence and, at the same time, should facilitate variation, so that the user does not get tired of watching the same story over and over again, as happens in a number of games.

TV is a medium which demands high responsiveness, that is, the satisfaction of users' expectations without compromising the quality of service. When watching TV one is not pleased, for instance, with an excess of interruptions during the presentation of a movie. In addition, programs are watched by a huge amount of people. Hence, adequate responsiveness requirements related to presentation flow and scalability have to be taken into account when we think about interactive TV storytelling.

In this paper, we describe a new model for controlling interactive TV storytelling processes which has been incorporated in the implementation of Logtell 2, a second version of the logic-based interactive storytelling system Logtell [Ciarlini et al 2005]. In the model, plot generation, user interaction and dramatization occur in parallel, and we strive to conciliate the requirements of different levels of interaction with coherence and diversity of stories, aiming at the high responsiveness demanded by the medium.

The paper is organized as follows. Section 2 gives a brief state of the art survey of storytelling systems and iTV. Section 3 presents the proposed model, and section 4 describes its implementation. Section 5 contains concluding remarks.

2. TV and Interactive Storytelling

Since the 1990's we have seen more and more a process of digital convergence, which has brought together many improvements and innovations in different areas, such as the infra-structure of communication networks, compression software and hardware, data transmission and broadcasting services [Furht 1996]. As a result of this convergence, new possibilities for TV broadcasting emerged such as open, satellite and cable Digital TV, Mobile TV, Web TV and IPTV. These developments have enhanced the experience of watching conventional TV. The first obvious benefit has been the improvement in image and sound quality, but changes have also been seen in the content that is available to spectators [Driscoll 2000]. Interactivity has scarcely begun to be explored, but it is already clear that it can radically change the way people watch TV. In particular, in developing countries, where TV has a much wider penetration than the Internet [CPqD 2005], interaction in an open digital TV environment has practical relevance, because it can significantly increase the access of a large part of the population to Education, Culture and Entertainment [Sancrini 2005].

Many opportunities for interactivity with users are possible, making it a topic of growing interest for research and development, both in industry and academia. There is, however, no consensus on how interaction with TV should happen. Possibilities of interaction depend on the computer power of the set-top boxes that receive and process the signal. Some specialists support the idea of *lazy interactivity*, with simple set-top boxes, by means of which the user has more limited options, but minimal effort and attention are demanded. Some others favor more powerful set-top boxes, but with easy and intuitive interfaces, so that users are able to watch TV and interact in a more active way. The specific features of set-top boxes that will prevail are not yet clear. It is quite possible however that there is space for both approaches. Anyway, it can be expected that the degree of interactivity will increase as interactive TV environments evolve. Interactive TV, or simply iTV, is a generic term that covers an ample set of possibilities of increasing sophistication, including:

- a weaker interactivity that corresponds to watching shows at a desired schedule, skipping ads, executing VCR commands, obtaining more information about what is shown (e.g. movies and news), etc.;
- directed and individualized advertising, together with sales and marketing;
- direct interaction with the presented content, changing, for instance, the ending of a story that is being watched; and
- the continuous interaction of a group of users with a shared content.

The first two items correspond to most of the new applications we have seen in recent years. The last two items are closely related to storytelling and tend to demand more research, but they have a strong appeal for the development of new applications that focus on Entertainment and/or Education. TV is a classical medium for telling stories in various formats, such as films, soap operas, cartoons and documentaries. Many possibilities for adapting these kinds of TV programs can be tried in order to incorporate interactivity in effective ways. Some experiments have already been made, but finding engaging formats that allow users to fully explore interactivity remains an open issue.

Some experiences of interactivity with TV content have already been carried out, even in conventional analog TV, although in an improvised and rigid way. An example is provided by reality shows, where spectators can make decisions by means of votes submitted by phone or via Internet.

Interaction with the content that is being presented is more complex than the other possibilities of interaction. It demands more sophisticated interaction methods and some kind of standardization of set-top boxes. Moreover, practical business concerns have to be addressed, because the user will have much more control on what is presented.

Projects like *ShapeShifting Media* [Ursu et al 2008] propose new forms of interactivity with TV content in opposition to the forms that have already been incorporated in interactive Digital TV environments. The project works with narrative models and some interesting applications have been developed as part of the project, such as *My News & Sports My Way*, in which the content of a continuous presentation of news is recombined in accordance with users' interest, and the romantic comedy *Accidental Lovers*, in which users can watch at real-time and influence a couple's relationship.

Despite the existence of some projects that tackle the interaction with TV content up to a certain extent, the dynamic creation of interactive stories at real-time for TV is still an open research issue. The mass production of coherent, diversified and engaging stories that can be influenced by users, in a comfortable way, is not a trivial task.

Interactive Storytelling has evolved as an interdisciplinary research area, involving Games, Filmmaking, Literature, Psychology, Cognitive Science and various fields of Computer Science, such as Computer Graphics and Artificial Intelligence.

Some approaches for Interactive Storytelling are classified as character-based [Cavazza 2002] because they focus on modeling characters as autonomous agents. In these approaches, stories emerge from the interaction between the characters. When this approach is adopted, it is easier to implement direct interaction with the characters, but harder to keep the story coherent. Other approaches are classified as plot-based [Grasbon and Brown 2001; Paiva et al 2001; Spierling et al 2002], since they focus on plot structure. They are directly influenced by the work of the Russian literary theoretician Vladimir Propp in his seminal work on the fairy-tales genre [Propp 1968]. In plot-based approaches, keeping the coherence of stories is easier, but opportunities of interaction are rather limited. Few attempts exist to combine both approach. Façade [Mateas and Stern 2003], for instance, keeps the characters' autonomy most of the time, but their goals and their behavior can be changed by a drama manager to move the plot forward. Genres that stress realism typically demand more coherence; on the other hand, in various genres, a free direct interaction between characters might result in a more engaging experience. In general, the right approach depends on the goal of each application and the genre of stories to be generated and told.

To create stories, a promising strategy is the use of automated planning algorithms, as in [Riedl and Young 2004], allowing to explore alternative ways whereby a logically connected chain of events could achieve the goals of the characters and/or those of the story. Diversity and coherence can be thus conciliated, but interaction must be constrained so as to limit the stories to those acceptable to the algorithms. In [Cavazza 2002], hierarchical task network (HTN) planning is used to control the way characters achieve their goals in accordance with user intervention. HTN planning tends to be efficient but less general, requiring the previous construction of a task hierarchy and methods to perform each task. In Façade, a reactive planning language is used to emulate the personality of believable agents. In Mimesis [Young 2001], a planner combining HTN and partial-order planning is used to create a storyline beforehand. Techniques of mediation are used at run time to guarantee coherence, including the adoption of alternative story lines or interventions for forcing the failure of users' actions.

Logtell is an interactive storytelling system based on logical modeling and planning-based simulation, which also tries to conciliate plot-based and character-based features. The main difference of the approach adopted in Logtell from other planning-based interactive storytelling systems is the goal of the system. Instead of working on alternatives for an entire story line, Logtell seeks to generate a maximum of different and coherent stories of a certain genre along multiple *simulation stages*, combined with user intervention. A formal model for capturing the logics of the story genre is specified to determine the scope of coherent alternatives. When plots are fully or partially generated, they can be dramatized via an animation of virtual actors in a 3D scenario.

Research on Interactive Storytelling may hopefully provide the basis for creating good models. The model proposed in this paper for interactive TV storytelling processes uses the original model of Logtell as a starting point, but also takes into account the specificities of a medium where high responsiveness is demanded and a majority of users may still prefer to assume a more passive behavior.

3. Proposed Model

The approach for interactive storytelling we have adopted assumes a third person viewpoint. We also assume that the conventions of the story genre can be logically modeled. The basic idea is to let the user interact with the story as if he or she were a "deus-ex-machina", with the power to choose alternatives for the future, cause the story to backtrack to previous points and try to force the occurrence of events and situations. User interventions have however to preserve coherence with the logical model. Interventions that do not make sense are rejected, but those that are found to be logically compatible can be incorporated, generating consequences to the rest of the story. In this way, the approach can be seen as an extension of the experience of watching a film on TV.

Interactions can vary from a level in which the user just watches a story as in conventional TV to a level of strong interventions in which the user is enabled to explore the possibilities allowed by the story genre. Although this ability was already provided by the first version of Logtell, the system still remained essentially a tool for logically modeling and simulating a story world obeying the rules of a genre. Varied and coherent plots could be generated with user intervention, but dramatization occurred only after the generation of the plot and there was no user intervention during the dramatization. The model proposed here uses the original model of Logtell as a starting point but modifies it in several ways to make it compatible with an iTV context. In order to do that, the model seeks to fulfill the following requirements:

I. It should be possible to create diverse stories, all of them coherent and resulting from interactions with the users.
II. Presentation flow must be continuous, that is, plot generation, user interaction and dramatization should occur in parallel without delays.
III. Comfortable and simple interaction methods at various levels should be provided so that different kinds of users can enjoy the experience.
IV. Users should be allowed both to interact with stories as single users and to share the control of stories with other users.
V. The underlying architecture should be scalable, in view of the massive nature of the medium.

In this section we present the basic architecture proposed by the model and then we discuss how the requirements listed above can be fulfilled by a system implementing this architecture.

3.1 Architecture

Figure 1 presents the client-server architecture proposed by the model. The client-side is responsible for user interaction and dramatization of stories. At the application server side there is a pool of servers sharing the responsibility of creating and controlling multiple stories, which are presented in different clients. This takes care of the case wherein multiple users are simultaneously sharing the same story. If clients are set-top boxes for interactive TV, their computational resources are limited, making it hard to perform CPU-intensive tasks such as automated planning. By concentrating simulation tasks in application servers, it is easier to achieve higher scalability. In addition, it is also easier to exert control when a single story is shared by many users.

The access of all modules to the context of the stories, specified in the Context Database, is always performed via the Context Control Module (CCM), which runs in the server. The context contains the description of the genre according to which stories are to be generated, and also the intended initial state specifying characters and the environment at the beginning of the story. The genre is basically described by: (a) a set of parameterized basic operations, with pre- and post-conditions, logically specifying the predetermined repertoire of events that can occur; (b) a set of goal-inference rules, specified in a temporal modal logic formalism, which define situations that lead characters to pursue the achievement of goals; (c) a library of typical plans, corresponding to typical combinations of operations for the achievement of specific goals, which is organized in "part-of" and "is-a" hierarchies; (d) logical descriptions of initial situations for the stories, introducing characters and their current properties; (e) a nondeterminisc automaton for each operation, specifying alternative ways whereby the event associated with the operation can be dramatized; and (f) graphical models of 3D virtual actors.

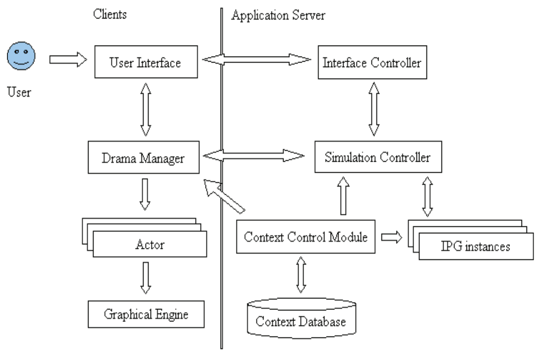

Fig. 1. Basic Architecture for iTV Storytelling

Plot generation is performed by the Interactive Plot Generator (IPG) [Ciarlini and Furtado, 2002]. IPG generates plots as a sequence of chapters. Each chapter corresponds to a cycle in which, subject to user interference, goals are inferred and planning is used to achieve the goals. IPG is controlled by the Simulation Controller. Multiple stages, each one corresponding to a chapter, usually occur in order to generate a plot. In case there is no user intervention, goals are inferred and events are planned continuously. Logical coherence of a requested user intervention is always checked before being incorporated, or else discarded if inconsistent.

The Simulation Controller is responsible for informing the Drama Manager, at the client side, the next events to be dramatized; receiving interaction requests and incorporating them in the story; selecting viable and hopefully interesting suggestions for users who are intent on performing strong interactions; controlling a number of instances of the Interactive Plot Generator, in order to obtain the next events to be dramatized; and controlling the time spent during simulation.

The Simulation Controller also keeps snapshots of the state of the simulation after the generation of each chapter, so that simulation can be resumed from any previous chapter of the story. In this way, intervention can be used to force a story to return to a previous point, from which alternative continuations can be tried. Snapshots are also important to allow different servers from a pool to deal with the same story, thus enhancing scalability. When the simulation of a story in about to be continued, the previous snapshot can be recovered from the database and the process can be resumed by any server available.

On the client side, the user interacts with the system via the User Interface, which informs the desired interactions to the Interface Controller placed at the server side. The Drama Manager requests the next event to be dramatized from the Simulation Controller, and controls actor instances for each character in a 3D environment running on the Graphical Engine. On the server side, the Interface Controller centralizes suggestions made by the various clients. When multiple users share the same story, interactions are selected according to the number of clients that requested them. When there is only one client, suggestions are automatically sent to the Simulation Controller.

The architecture in Figure 1 avoids the transmission of video on demand during the storytelling process, a precaution that ought to be taken in order to allow many simultaneous stories without compromising bandwidth. All data necessary for 3D dramatization on the client side can be transmitted before starting the simulation. During the process, the transmitted data is restricted to information about user interaction, the indication of events to be dramatized and synchronization commands. This strategy assumes however that there is enough computational power at the client side to generate a 3D animation. In another scenario, with many users sharing a limited number of stories but having very little computational power, the architecture can be modified to have a Drama Manager for each story running on the server-side. Each Drama Manager would then generate video to be broadcasted to the users that share the corresponding story.

In the sequel, we discuss the main strategies to control interactive storytelling processes in the described architecture.

3.2 Coherence and Diversity of Stories

Coherence and diversity of narratives are key factors to the success of any interactive storytelling application for iTV. If stories do not seem plausible and coherent with respect to the genre, the user may lose interest for the experience. If generated stories have little variation, their ability to entertain and surprise tends also to be reduced. After a few trials, users would be ready to discard the application.

Balancing coherence and diversity of stories with interactivity is a difficult challenge. Too much interactivity can easily hinder the coherence of the story. Too little interactivity on the other hand would reduce the variation in stories and the impact of the experience.

Conciliation of coherence, diversity and interactivity can be achieved by means of hard-coding various coherent alternatives. Façade, for instance, as previously mentioned, is among the most successful interactive storytelling applications, allowing users to enjoy an interesting interactive experience for about 20 minutes. The structure of the system demanded however a huge authorial effort (with thousands of lines of code) to model a single dramatic situation with different possible outcomes. Another problem with this kind of solution is that authors lacking the required kind of programming expertise may find difficult to directly model situations as part of the application code. For the mass production of interactive storytelling contexts, as would be necessary for iTV, other solutions have to be sought.

The strategy that we propose is the construction of a logical model for the genre and the use of planning and inference of goals to guarantee coherence while exploring diversity. Plot generation starts by inferring goals for the various characters (and for the story as a whole) from the initial situation. Given this initial input, the system uses a planner that inserts events in the story-plot in order to fulfill the goals. When the planner detects that all goals have been either achieved or abandoned, the first chapter of the story is finished. If the user does not like the story, IPG can be asked to generate a different alternative for a chapter and to develop the story from this point on. If the user does not interfere in the process, chapters are continuously generated by inferring new goals from the situations generated in the previous chapter. If new goals are inferred, the planner is activated again to fulfill them.

The process thus alternates goal-inference and planning until the moment the user decides to stop or no new goal is inferred. Users can also interfere in the process by choosing alternatives and forcing the occurrence of events and situations as described in section 3.4. Notice that, in this process, we mix forward and backward reasoning. In the goal-inference phase, we adopt forward reasoning: past situations generate goals to be fulfilled in the future. In the planning phase, an event inserted in the plot for the achievement of a goal may have unsatisfied pre-conditions, to be handled through backward reasoning. To establish a pre-condition, the planner can insert previous events with further unfulfilled pre-conditions, and so on. The planner used in IPG is a partial-order planner, adopting a least-commitment strategy to more easily accomplish the conciliation of different goals. Constraints (including the order of events) are established only when necessary, and all possibilities for solving conflicts between events in the establishment of pre-conditions are considered.

IPG provides a base for virtually creating any plot compatible with the rules of the genre. At each stage, the user can reject the alternative being currently presented and ask for another, or may opt for a direct intervention. And whenever the user intervention is compatible with the genre, IPG provides means for adapting the story so that the user's contribution can be incorporated. In this way, the adopted approach aims to provide coherence and diversity by construction. Authorial effort is still necessary to formally model the interactive storytelling context, but this is inevitable in the creation of any interactive or conventional story. The difference is that, thanks to the plan-based support described here, plots need not be devised beforehand by the author.

3.3. Continuous Presentation Flow

When spectators watch movies on TV, events are presented continuously. This is not a difficult task because the whole story is generated and filmed beforehand. In iTV this is not the case, but, if iTV storytelling purports to be an extension of the experience of watching conventional films, the presentation flow should likewise be continuous.

A premise of our model is that users should feel that they can change, to a significant extent, the story being presented according to their will. The continuous presentation of a story that is modified by user interventions is a challenge, even if dramatization is performed by means of 3D animations instead of real actors. If we want to reach the same level of coherence of a conventional story, plots have to be continuously adapted to incorporate user interventions. Checking the coherence of an intervention at real-time and computing the possible consequences of the intervention to the rest of the story is not trivial and may become excessively time-consuming. Some kind of synchronization between plot generation, user interaction and dramatization is then mandatory.

In order to synchronize the processes, narratives are divided into chapters. While a chapter is being presented to the user, IPG can already start generating the future chapters. When user interventions are coherent, they are incorporated in the next chapter. In this way, we try to keep plot generation some steps ahead of the dramatization, so that chapters are continuously generated and dramatized. The main problem occurs when a user intervenes in the story, trying to force a situation or the occurrence of an event. Since user interaction affects the situation of the chapter currently being presented, future chapters previously generated without taking the intervention into consideration would no longer be useful. The difficulty is that, as the next chapter has to be ready before the end of the dramatization of the current chapter, there is a risk of interruption in the presentation flow. The following strategy is applied, with two options. When the Simulation Controller detects that more time is needed for generating the next chapter, a message is sent to the Drama Manager to the effect that the duration of the remaining events in the current chapter will be extended, as detailed in [Doria et al 2008]. If there is no way to extend the chapter being presented until the next chapter is ready, the user intervention is discarded, as if it were inconsistent. In this case, the chapter that had been generated without incorporating the user intervention is used.

An alternative to reduce the number of times when coherent user interventions are rejected is the use of other instances generated by IPG, besides those corresponding to the current flow of the stories. Such instances can be used to try to anticipate the effects of possible user interventions, so that future chapters will be ready when necessary. In a continuous presentation of the story, strong user interventions are based on suggestions given by the system. The Simulation Controller is then aware of possible user interventions, and IPG instances that incorporate each one of them can be started.

Due to the combinatorial complexity of automatic planning, the most time-consuming part of the simulation corresponds to planning events to reach inferred goals or goals imposed by the user. In the simple scenario used to test our model, the current planning algorithm adopted by IPG has been able to generate chapters in due time. However, in order to maintain responsiveness in more complex scenarios, it may be necessary to enhance the performance of our planner. The possibilities envisaged include combining the current planner with graph planning techniques [Geverini and Serina 2002], and resorting to heuristics and control strategies [Hoffman 2001] and/or HTN techniques [Nau et al 2003].

3.4. Interaction Methods

In an iTV storytelling application, the user should be able to easily interact with the story. Interaction must never disrupt the user's immersion in the story, exactly as one expects from conventional TV. In opposition to what occurs in various games, the user's ability to quickly react is not so critical, because the effort involved in interacting with a story is basically intellectual, rather than physical.

Interaction methods have also to take into consideration the different kinds of prospective users of the application. Some users may want to essentially remain as spectators, willing to interact very little with the story. Others would be prepared to continuously intervene in the story, actively determining the way the plot unfolds. It is then necessary to provide more than one method, to accommodate different levels of intervention in the stories.

Our model offers the possibility of both weak and strong interventions in the story. By means of weak interventions, the user can select alternatives that are automatically generated by IPG. Strong interventions are used to try to force the occurrence of events or specific situations. The window in Figure 2 shows the current version of our tool's interface, through which users interact with the story being dramatized as displayed in the main window. The interface has certainly to be improved and adapted to devices other than a desktop computer, but has already served to check the viability of our initial set of interaction mechanisms.

Chapters are continuously generated and presented in the main window. When a chapter is being presented, a new line corresponding to that chapter is inserted into the list box *Chapters*. The description in natural language of a selected chapter appears in a text box. Weak interventions occur by means of the commands *Rewind* and *Another*. In order to execute such commands, the user has only to select a chapter and press the corresponding button.

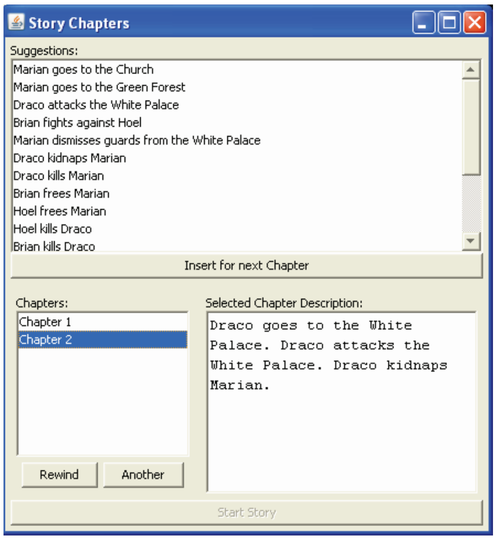

Figure 2: Window for continuous interaction.

The *Rewind* command was so named by analogy with the rewinding fuction of Video Cassette Recorders, but, in fact, it is considerably more powerful. By executing this command, we allow the user to "return" until the time the selected chapter was being presented. The chapter is presented again and the user has the opportunity of interacting with the system and checking alternatives for the following chapters. When this command is executed, the Simulation Controller retrieves the snapshot corresponding to the storytelling process at the time the chapter was presented and resumes the simulation from this point, discarding the snapshots of the next states, which will be generated again in accordance with the user's interactions.

Command *Another* is used to ask the system to provide an alternative for the selected chapter. It is similar to command *Rewind*, in that it also involves a return until the time the selected chapter was being presented. In response to the command, IPG generates another solution for the goals that were reached. In this way a different combination of events can be generated for the chapter, whereby a completely different continuation of the story can be developed.

Commands *Rewind* and *Another* demand a moderate mental effort from the user and are quite intuitive. They are useful when users want to explore other alternatives for the story without having to assume a more active participation in the plot generation process.

In contrast, strong interventions correspond to the specification of events and situations that should occur in the next chapter. Situations are considered as goals to be achieved at a certain time, and events can have unfulfilled preconditions that might demand the insertion of more events. In such cases, IPG has to plan a chapter with additional events and constraints that make the user intervention consistent with the plot and the rules of the genre. If this is not possible, the user intervention is simply rejected. In a continuous presentation, the specification of events and situations from scratch might impose an excessive burden to the user. To make the process simpler, the model proposes a mechanism in which viable strong interventions are *suggested* to the users, so that they can simply select the one that better suits their taste and press a button. The list box *Suggestions* in the interface contains suggestions from which the user can select at a given time. The list is updated whenever the presentation of a new chapter starts, so that only meaningful suggestions are presented.

The Simulation Controller is responsible for generating suggestions of strong interventions. Suggestions should be consistent and lead the plot towards different outcomes. The methods below are proposed by the model to obtain meaningful suggestions:

- The first method corresponds to the specification by the author of rules for the inference of suggestions. Such rules are specified in the same temporal modal logic used to describe goal-inference rules. The Simulation Controller evaluates all rules in accordance with the context of the current chapter and collects suggestions that would make sense. Suggestions obtained in this way can be quite helpful to create an interesting set of options for the users. This method demands however an additional authorial effort.
- A second method uses a library of typical plans organized in a hierarchy of events. Typical plans usually consist of certain combinations of events whereby the various characters pursue their goals, but they can also correspond to *motifs*, i.e. recurring structures compiled in the course of critical studies on the genre [Polti 1945]. IPG contains a procedure for the recognition of plans, based on an algorithm specified by Kautz [Kautz 1991]. The procedure is able to discover that some given events are compatible with a motif for which we have a typical plan, enabling the Simulation Controller to suggest the inclusion of additional events contained in the plan.
- A third method corresponds to an analysis of goal-inference rules in face of the current plot. If the system detects that a goal-inference rule will be triggered if a certain situation is verified in the next chapter, this situation can be selected as an interesting suggestion.

Interactive TV is still a novel environment. A model for iTV storytelling has then to be open enough to incorporate new kinds of interaction. With this in mind, the model considers the possibility of incorporating other methods for weak and strong interventions, such as letting users insert abstract events and situations in the story, which are automatically specialized by the simulation process; tune narrative tensions by means of numeric scales referring to levels of violence, romantic turns, etc.; and communicate with the system by entering phrases in (a restricted subset of) natural language.

Besides the support for real-time generation and dramatization of stories, it is important to support the authorial effort. The original version of Logtell already worked in a step-by-step mode, in which the partial plot after each chapter was presented as a graph and could be inspected in detail. In this mode, weak and strong interventions are also possible and dramatization for the plot generated so far has to be explicitly activated by the user. The user could even analyze the generation of the whole plot and activate dramatization only at the end. The present model considers that the continuous interaction mode previously presented should coexist with the step-by-step mode.

3.5. Sharing Stories

A most important iTV storytelling mode of application is the creation of stories that are influenced by a great number of users. Weak interventions, via commands like *Rewind* and *Another*, are not so appealing when multiple users are watching the same story. More attention should then be directed to interaction methods enabling strong interventions.

Criteria on how to deal with different interventions are necessary. A first possibility is to consider that, when users ask the system to incorporate a specific suggestion, they are only voting for the suggestion. The most voted suggestion would then be chosen to be incorporated.

The Interface Controller organizes the interaction with clients and interacts with the Simulation Controller as if there were a single user. In order to do that, it coordinates the simultaneous dramatization and the presentation of suggestions for strong interventions in the various clients. It also controls the time during which users' choices are considered. After computing the most voted suggestion, the Interface Controller checks whether the number of votes reaches a minimum threshold. If this is the case, the suggestion is sent to the Simulation Controller. However, the selection of strong interventions does not have to be limited to considering only the most voted intervention. Other possible strategies are:

- The number of votes can be weighted by the potential of each option to trigger goal-inference rules. In this way, options that generate more interesting situations tend to be chosen.
- Compatible interventions can be combined in the same chapter. In particular, different groups of users may have different options. They can, for instance, be distinguished by the characters the group components decide to support. IPG would then try to combine the choices of all groups.

When the possibility of influencing the story by tuning narrative tensions is admitted, the numeric scales can be controlled by the average of the values assigned by the users.

The way groups are formed to share a story is also an issue to be resolved as part of the implementation of the model. A simple solution is to assume that any user is allowed to schedule the start of a story based on a specific context at a certain time. As other users notice one such story in a list of scheduled stories, they may then be tempted to join the group. Users can either have equal rights to intervene in the story or not; in the latter case, different criteria can be established to assign their rights and privileges. Methods for the communication among users who share the same story can also be devised, so that they would be able to discuss their interventions.

Other alternative methods and criteria for multiuser interaction can be examined. The model proposes a platform to experiment with them in order to test their feasibility, and determine which ones are more engaging for the audience.

3.6. Scalability

Due the massive aspect of the medium, an iTV storytelling system has to cope with the possibility that a huge number of people might be using the application at the same time. They can use the system simultaneously to generate stories in different contexts; they can share a same context to obtain different stories; and they can simply share stories. As TV spectators are not used to experiment unexpected delays, the scalability of an iTV storytelling application is crucial.

In our model, plot generation consumes considerable computational resources in terms of CPU time and memory. In order to avoid the creation of a bottleneck, IPG and the Simulation Controller can run in multiple servers. Snapshots of each story can be temporarily stored in a database, so that any available server can restore a specific snapshot and generate the next chapter for the corresponding story. In this way, responsiveness can be maintained by simply increasing the number of servers in the pool. Additional servers can also be considered to avoid bottlenecks in the access to the database and in the communication with clients.

Regarding scalability, another important issue to be taken into account is that users might want to interact with the application via different platforms and with different local computational resources. In particular, if local resources are very limited, the execution of the dramatization in the server tends to be mandatory. For this purpose, it may be necessary to adopt a hybrid architecture in which some clients can dramatize their own stories locally, and other clients can only share stories dramatized and broadcasted from the server.

4. Model Implementation

In this section, we explain the main issues and the options adopted for the implementation of Logtell 2, a new version of Logtell that incorporates the iTV storytelling model described in section 3.

4.1 Application Environment

Logtell 2 was built utilizing a modular architecture, employing different technologies appropriate to its intended functionalities. The User Interface, the Interface Controller and the Simulation Controller modules were implemented in Java. The Drama Manager maintains the original 3D engine implemented in the first version of Logtell, coded in C++. The Drama Manager communicates with the parts of the system implemented in Java via a DLL that provides an interface connection with the C++ code. The IPG planner is implemented in a version of SICStus Prolog supporting Constraint Programming. The Simulation Controller manages plot generation by accessing IPG via a Java bean that uses native C++ calls to communicate with the SICstus Prolog interpreter.

One of the main requirements for the implementation of Logtell 2 was the use of a client-server environment designed for availability and scalability. Since the application is mostly Java-based, given that both the interface and server side code is implemented in this language, a popular J2EE application server was used, namely JBoss [Marrs and Davis 2005]. JBoss provides a good set of facilities such as distributed services, security, support for asynchronous messages, remote proxies, database access, Web Services and HTTP servers, all desirable for a complete system for interactive TV. Regarding scalability in particular, JBoss makes easier the construction of a pool of servers to provide services to a great number of clients.

It is also important to notice that, since the JBoss architecture handles Web Services, different ways of accessing Logtell 2 can be provided. Since all services were codded using the Enterprise Java Beans 3.0 standard, under the form of Stateless Session Beans, their conversion into Web Services becomes practically automatic, thus making it possible to use mobiles among other ways of access.

4.2 Logtell 2's Modules

The effort for constructing a distributed iTV storytelling processor mainly involved the implementation of the User Interface, the Interface Controller and the Simulation Controller modules.

The services provided by the application servers in Logtell 2 follow the EJB standard. The Simulation Controller was implemented as a Stateless Session Bean. The logic control tasks for the creation of stories have been assigned to distinct submodules: generic services to manipulate the story are handled by StoryManagerService, while the generation of the story-plot via IPG is performed by subclasses of the AbstractStoryWriter: one for continuous stories (ContinuousStoryWriter) and one for stories generated in step-by-step mode (StepByStepStoryWriter)

The StoryManagerService centralizes the services of updating and retrieving stories and their respective chapter snapshots, using a database abstraction in the form of Database Abstract Objects (DAOs). Stories are iteratively generated. Whenever a new simulation cycle is requested, the service restores the previous story snapshot and then proceeds to write another part of the story. In continuous mode, the code that prompts story generation organizes the plot composition in chapters, wherein the total order of the events is established automatically, obeying partial-order constraints established by IPG. In step-by-step mode, the ordering is determined by the user on the graphic interface.

In the continuous mode, there is an instance of ContinuousStoryWriter for each story that is being generated on the fly. The ContinuousStoryWriter works as if it were a “live screenwriter”. For that, it manipulates a StepByStepStoryWriter instance, keeping control over the story that is being generated. The ContinuousStoryWriter implements the strategy of always being ahead of what is being watched by (one or more) users. When, in the continuous mode, a chapter is requested by the client, the server side checks whether this is the most recently created. In this case, to avoid an interruption to the story flow, a message is sent to the corresponding ContinuousStoryWriter of that story, requesting the generation of the next chapter.

The Simulation Controller is also responsible for providing the new forms of strong intervention in continuous mode, which are based on suggestions of strong interventions. Strong interventions are only incorporated if IPG validates them as coherent. In addition, the chapter incorporating the intervention has to be generated before the end of the dramatization in all clients that are watching the same story.

The multi-user story generation process is very similar to regular continuous mode. The main difference is that multi-user stories do not start instantaneously. They are scheduled to start at a certain time by one user. At any time, a user can inspect the set of scheduled stories and join the group of users that will watch and interact with the story selected. At the scheduled starting time, the story dramatizations start in all clients and are synchronized from this moment on. For the time being, strong interactions provided by different clients are chosen to be incorporated only on the basis of the most-voted strategy.

The User Interface contains submodules that implement the interface with users in continuous mode and in step-by-step mode. In step-by-step mode, the User Interface communicates directly with the Simulation Controller. In continuous mode, it communicates with the Interface Controller. The Interface Controller is in charge of centralizing the interaction (in case of multi-user interaction), redirecting user interventions to the Simulation Controller and synchronizing the list of suggestions of strong interventions in the clients.

In the current version of Logtell 2, the Drama Manager and the IPG planner have received relatively minor modifications with respect to their original versions. In IPG, modifications were introduced to allow the Simulation Controller to save and recover story snapshots after the generation of each chapter. The evaluation of rules for inferring promising strong interventions was the other extension of IPG incorporated in this version. The Drama Manager remains essentially the same module of Logtell’s first version, but another version is presently under development, which uses a nondeterministic dramatization model to allow the system to control the duration of events and provide different dramatizations for the same event [Doria et al 2008]. In order to provide better quality in 3D animations, the current 3D engine is also being replaced by a module that controls characters in the UNITY 3D game engine.

4.3 Using the Prototype

In order to evaluate the prototype, we utilized the same storytelling context adopted in [Ciarlini et al 2005] where the first version of Logtell was described, with minor modifications. The context corresponds to a small sub-class of the popular *Swords and Dragons* genre. In this context, the events that can occur correspond to *attacks to the opponent's home, fights between characters, kidnapping of a victim, liberation of a victim, charms, weddings*, etc. Goal-inference rules used in this context establish relations between situations and goals, such as: *if the villain is strong, the hero wants to become even stronger*; *if a victim is kidnapped, a hero will want to rescue her*; etc.

The prototype was applied to generate stories in this context, using machines connected via a local network. When stories were generated and dramatized in continuous mode, without user intervention, there was no perceptible interruption (less than 200 ms) between the time a chapter is over on the client and the time the following chapter starts. Chapters were generated on the server with more than enough time left while the user was watching the previous chapters. In situations where network problems could arise, there might be a chance that the user's experience would be affected; fortunately this is not likely to happen, since the amount of information that is sent to the client is very small in the current way the system is designed.

Tests were also performed using the forms of interaction specific to the continuous mode. When executing the command *Rewind*, the time to resume the story at the indicated chapter was also insignificant (~ 350 ms). The execution of command *Another* takes a little more time (~ 5 seconds), because it demands not only the recovery of a previous context, but also the generation of another solution for the chapter, but the time consumed still seemed quite acceptable. In general, it was verified that, under ideal conditions, the results were satisfactory for the continuous presentation flow when using weak interventions. By "ideal conditions" we mean a situation where the computational resources in the application environment are not overloaded.

When using strong interventions, there were also no additional delays compared to the tests where the user watched passively. This happens because, in continuous mode, interventions are incorporated only when the server still has time to prepare the next chapter. In the worst case, the intervention is ignored but no interruption occurs. In most occasions, the dramatization time of a chapter showed to be long enough to allow the incorporation of coherent interventions.

Regarding the diversity of the stories, it was possible to obtain in continuous mode most of the stories that a user could obtain in step-by-step mode. The presentation of meaningful suggestions for strong interventions showed to be effective to allow users to actively intervene in the story with minimal effort.

We had no difficulty to generate different stories simultaneously. The process of sharing stories worked equally well in the same circumstances. Tests were done with clients running both on the same and on different machines. As far as scalability is concerned, more tests have still to be performed, but preliminary results indicate that the model can work well with a large number of clients.

5. Concluding Remarks

The model of interactive storytelling, discussed in the present paper, aims at the generation and dramatization, by means of 3D animations, of interactive stories in iTV environments. The implementation and use of the Logtell 2 prototype, based on the model, served to confirm that the architecture and methods proposed are viable and able to cope with the requirements of coherence and diversity of story-plots, continuous presentation flow, comfort and ease in interaction, multiple user participation in stories, and scalability. The prototype currently runs on a local network. We are now preparing a version for the Web, with the ultimate purpose of reaching the open Brazilian Digital TV environment. Tests on an increasingly larger scale will be performed at each stage of the implementation.

In more detail, we are working on a new version of the Drama Manager with better 3D animations, additional dramatization possibilities, and with the ability to control the duration of each event. Another short-term objective of the project is the full implementation of features of the model for which a simplified solution was initially adopted. For instance, the coordination of a pool of servers will supersede the limited use of a single server to handle all stories. Also bulkier snapshots should no longer be kept in main memory, being transferred to database storage. More advanced methods for user interaction and evaluation of suggestions for strong interventions are still to be implemented. As extensions to the model itself, we are considering the use of nondeterministic planners for plot generation, and of frame-based schemes and methods to deal with the drives, attitudes and emotions of the acting characters.

Acknowledgements

The authors are grateful to Fundação CAPES for partially supporting this work.

References

CAVAZZA, M., CHARLES, F. AND MEAD, S. 2002. Character-based interactive storytelling. IEEE Intel-ligent Systems, special issue on AI in Interactive Entertainment, 17(4):17-24

CIARLINI, A.E.M., POZZER, C. T., FURTADO AND A.L., FEIJO, B., 2005. A Logic-Based Tool for Interactive Generation and Dramatization of Stories. In: Proc. ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Valencia.

CIARLINI, A. AND FURTADO, A. 2002. Understanding and Simulating Narratives in the Context of Information Systems. In Proc. ER'2002 – 21st. International Conference on Conceptual Modelling, Tampere, Finland, Oct. 2002

CPQD. 2008. Modelo de Referência – Sistema Brasileiro de Televisão Digital Terrestre. Vers. PD.30.12.36A.0002A/RT-08-AB. 2006. <http://www.tvdi.inf.br/index.php?s=artigos>.

DORIA, T. R., CIARLINI, A. E. M. AND ANDREATTA, A., 2008. A Nondeterministic Model for Controlling the Dramatization of Interactive Stories. In: Proceedings of the ACM MM2008 - 2nd ACM Workshop on Story Representation, Mechanism and Context - SRMC'08.

DRISCOLL, G. 2000. The essential guide to digital set-top boxes and interactive TV. Prentice Hall, Upper Saddle River, NJ, 1st edition, Nov. 2000.

FURHT, B. 1996. Interactive television systems. In: Proc.1996 ACM Symposium on Applied Computing, p. 7-11, Philadelphia, Pennsylvania, February 1996. ACM Press.

GRASBON, D. AND BRAUN, N. 2001. A morphological approach to interactive storytelling. In: Proc. CAST01, Living in Mixed Realities. Special issue of Netzspannung.org/journal, the Magazine for Media Production and Intermedia Research, pp. 337-340, Sankt Augustin, Germany

GEVERINI, A. AND SERINA, I. 2002. LPG: A planner based on local search for planning graphs. In Proceedings of the International Conference on AI Planning Systems (AIPS), pp. 968-973.

HOFFMAN, J. 2001. FF: The Fast-Forward planning system. AI Magazine, v. 22, n. 3, pp. 57-62.

KAUTZ, H. A. 1991. A Formal Theory of Plan Recognition and its Implementation. In: Allen, J. F. et al (eds.): Reasoning about Plans. Morgan Kaufmann, San Mateo

MARRS, T. AND DAVIS, S. 2005. JBoss at Work: A Practical Guide. O'Reilly Media, Inc.

MATEAS, M. AND STERN, A. 2003. Façade: An Experiment in Building a Fully-Realized Interactive Drama. In: Proc. Game Developers Conference, p. 4-8, 2003.

NAU, D. S., AU, T.-C., ILGHAMI, O., KUTER, U., MURDOCK, W., WU, D. AND YAMAN, F. 2003. SHOP2: An HTN planning system. Journal of Artificial Intelligence Research, 20:379-404.

PAIVA, A., MACHADO, I. AND PRADA, R. 2001. Heroes, villains, magicians, ... Dramatis personae in a virtual story creation environment. In: Proc. Intelligent User Interfaces

POLTI, G. 1945. Thirty-Six Dramatic Situations. Whitefish, MT: Kessinger Publishing.

PROPP, V. 1968 Morphology of the Folktale, Laurence Scott (trans.), Austin: University of Texas Press.

RIEDL, M. AND YOUNG, R.M. 2004. An intent-driven planner for multi-agent story generation. In 3rd International Conference on Autonomous Agents and Multi Agent Systems, New York, 186-193

SANCRINI, M., 2005. O Uso da Televisão Digital no Contexto Educativo, Educação Temática Digital, Campinas, v. 7, n. 1, ISSN 1676-2592, pp. 31-44.

SPIERLING, U., BRAUN, N., IURGEL, I. AND GRASBON, D. 2002. Setting the scene: playing digital director in interactive storytelling and creation. Computers & Graphics, v. 26, n.1, pp. 31-44, 2002.

URSU, M.F., THOMAS, M., KEGEL, I., et al. 2008. Interactive TV Narratives: Opportunities, Progress, and Challenges. In: ACM Transactions on Multimedia Computing, Communications, and Applications, v. 4, n. 4. ISSN:1551-6857

YOUNG, R. 2001. An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment. The Working Notes of the AAAI Spring Symp. on Artificial Intelligence and Interactive Entertainment