1 Introduction

Interactive storytelling as a form of massive digital entertainment has many open questions. How can we transform a TV program into a skilled storyteller capable of adapting the reactions of hundred thousand spectators simultaneously? How can we convert the audience into co-authors of a story in such a way that the evolving plot surprises even the original author of the script? These questions drive the attention of both TV producers and researchers. Since the 2000’s, research works on interactive storytelling have been growing at a fast pace. However, the literature on interactive storytelling for digital television is still scarce, especially when we are looking for systems that can deal with a great variety of devices. Furthermore, most of the systems are based on a predetermined and fixed tree of possibilities. In the present thesis, we propose a model and an architecture that address some of the above-mentioned issues.

In this thesis the model used in a previous Logtell interactive storytelling system [1],[3] is adapted to a new massive multi-user system. In the Logtell system, stories evolve in chapters and each chapter is a sequence of logically consistent events. Spectators interact with the story by making suggestions for the next chapter that will be accepted if coherence with the mathematical logic model of the story is preserved. Fig. 1 illustrates Logtell’s evolutionary cycle of a story composed by two processing stages: plot generation (creating and managing the story itself) and dramatization (that is, the actual story being shown based on its plot). A first draft of this story cycle was presented in a previous work by the author [5] and a more complete version was recently implemented for the present thesis. In the plot generation stage, a plan π (which is a sequence of events in a story to be dramatized) and a list of suggestions for the next chapter are generated from the initial state s_0. The initial state is a set of rules and facts that define genre, characters, and general situations. The user can pick up a suggestion (a possible interaction) from the list or write a new one. The execution of the events e^i

PUC-Rio - Certificação Digital Nº 0912917/CA

produces a set of new facts to be added (Facts$^+$) and a set of facts to be excluded (Facts$^-$):

$$s0 = (s0 \cup Facts+) -> Facts-$$

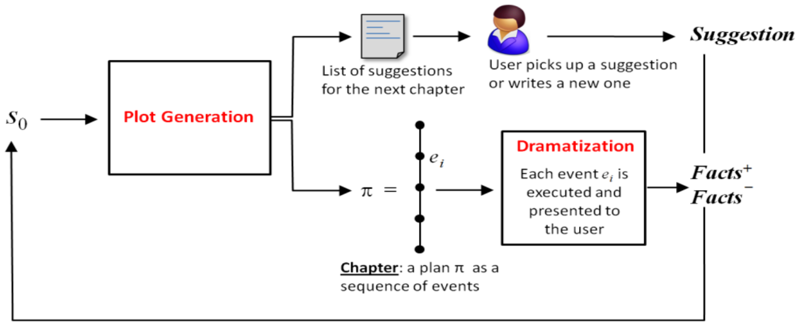
Suggestion
List of suggestions for the next chapter
User picks up a suggestion or writes a new one
s_0
Plot Generation
e_i
π =
Dramatization
Each event e_i is executed and presented to the user
$Facts^+$
$Facts^-$
Chapter: a plan π as a sequence of events

Figure 1 Story Cicle

The list of suggestions is created by the same logical inference mechanism that generates the plan π. Therefore, the evolution of a story is always a surprise even for the original author. In the Logtell model, the inference mechanism is based upon a temporal modal logic framework [24] and a partial order planning algorithm. The inference mechanism is written in a constraint logic programming language [4] and details of a single-user system can be found in previous works [1][3]. The prototype presented in this thesis implements the same small sub-class of the popular Swords and Dragons genre used by those previous works, telling a simple story about damsels, heroes, villains, wizards, courage, revenge, and love. The logical rules of the story are also the same found in previous works [1][3][5].

Our planning algorithm considers the following rules and facts that define the genre and the initial conditions of the story: (a) a set of goal-inference rules, which define situations that lead characters to pursue the achievements of goals (e.g., if a damsel (i.e. a victim) is kidnapped, a hero will want to rescue her); (b) a set of parameterized basic operators, with pre- and post-conditions (e.g. the operator go(Character, Place), which indicates a character moving to a specific place, has simple preconditions, such as Character is alive and Character is not kidnapped by someone, and obvious post conditions, such as Place is the new location of Character); (c) logical descriptions of initial situations of the stories, introducing characters, their relationships, and their current properties (e.g., Hoel

PUC-Rio - Certificação Digital Nº 0912917/CA

PUC-Rio - Certificação Digital Nº 0912917/CA

is a hero, Brian is a hero, Marian is a princess, Draco is a villain, and Draco is a Dragon).

A previous work [5] proposed a client-server model with rich clients (i.e. clients that provide rich functionality independent of the server). In that work, we assume that the spectators' devices could cope with heavy local processing. In the present thesis, we take an opposite strategy; that is, we search for a flexible and scalable architecture for interactive storytelling in digital television that allows thin clients and a great diversity of platforms, as illustrated in Fig.2. Interactive storytelling systems for digital television should consider the fact that PCs, smartphones, digital TV set-top boxes, and tablets have different capabilities to render graphics and run artificial intelligence tasks. Moreover, those platforms have distinct screen resolutions, performance, and user interfaces. In this thesis is shown a model of interactive storytelling for digital television that provides simple portability for different environments without losing the richness of interactivity and narrative coherence. We believe that the model we present in this thesis can contribute to create better solutions for broadcasting interactive storytelling on digital television networks.

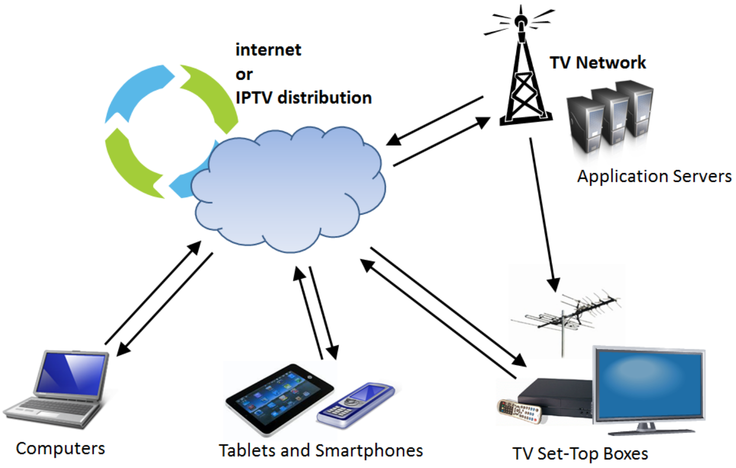

Figure 2 Network environment of digital television for massive multiplayer interactive storytelling systems

The main purpose of the presented model is to transfer the plot generation and dramatization tasks of the chapter generation cycle (Fig. 2) from the clients to a cluster of servers. In this model, we have video streams as responses to the

PUC-Rio - Certificação Digital Nº 0912917/CA

clients' requests. For this reason this proposal can be classified as a Stream-based Interactive Storytelling architecture. The proposed model can cope with most of the strategies for spectators sharing interactive stories. The model's architecture allows a light and highly responsive approach to web services by using HTTP as the transport layer, JSON [7] as the data layer, and REST [8] as architectural style. This approach is more flexible and lighter than the classic layers of XML as the data layer and SOAP as the protocol layer. Furthermore, the design of the architecture allows that different dramatization paradigms are easily adapted, such as text only, audio only (for visual impaired people), 3D rendering, video-based storytelling, 2D comics, 2D rendering, and even some combinations of them (e.g. text and 2D rendering). This approach to TV interactive storytelling allows a new form of massive digital entertainment, in which spectators can choose the type of dramatization they prefer.

This thesis presents a contribution towards a distributed architecture for interactive storytelling that considers a multitude of environments and interfaces. As far as the authors are aware, there is no work in the literature that contemplates massive multiplayer interactive storytelling based on temporal modal logic and offers interactive dramatization on a variety of devices. Works on interactive storytelling usually focus on a single spectator experience or, even in the case of the multiuser systems, on individualized experiences similar to multiplayer games, which stray from the interactive storytelling proposal. Also, another contribution are the voting systems presented, that use multiple strategies in order to support massive interaction and reaching collective decisions watching stories.

This thesis is organized as follows. Chapter 2 presents related works. The architecture for a stream-based interactive storytelling system and the proposed model of massive interaction is presented in Chapter 3. Processes and methods of this architecture, the interface of the prototypes and evaluation of results can be found in Chapter 4. Conclusions and future work are presented in Chapter 5.