2
Theoretical Background

In this chapter, some necessary concepts are presented in order to delimit the scope of this work. Also this chapter shows the main areas and questions related to storytelling and the most important issues that are necessary to support the requirements of the proposed model.

The main aspects presented in this chapter are: (1) Interactive Storytelling: what it is an interactive story system and how different systems handle stories and interactions; (2) Logtell: one of the most versatile systems to deal with massive interactive storytelling; (3) interactive television: how it works and how its concepts are related to the proposed model; and (4) voting: how difficult it is to manage multiple people intentions and what are the main ways to deal with them.

2.1.
Interactive Storytelling

Independent, non-academic projects, like The Written World [9], show that there is a demand for multiplayer interactive storytelling as a product. This project, which is a web-based product financed by a crowd-sourcing agency, proposes an interactive storytelling system focused on an interactive narrative presented in a text format, where a user plays the role of managing “the world” and the other is “the hero”.

Figure 3 Sleep is Death [10]

Another multi-user interactive storytelling system (Figure 3) is Sleep is Death [10], based on a two-player experience, where one is the author and the other the spectator. In this system, there is no backend to have an automated story generation – the author must prepare scenes in a graphical environment with interactive elements and the other user explores this scenario, similar to a “live” adventure game.

Even though these systems rely on user creativity to provide the interactive storytelling experience, and there isn't an automated story generation engine, they demonstrate that people are interested in sharing stories with other people. Storytelling is a tradition among cultures that is commonly related to a shared experience, with story tellers that amuse their audience.

Figure 4 Façade [11]

Figure 4 illustrates Façade [11], a game where the user plays a guest invited for drinks with a couple of friends. By using the mouse and by typing phrases, the user interacts with objects and talks to other game characters using natural language, being able to influence plot direction. In other words, the user may help the couple understand each other, break up, or even be expelled from their house. Façade is regarded as one of the most successful interactive storytelling systems. However, it is based on a single user experience, and only available for computers.

Riedl and co-authors [12] conducted experiments to find out how to make multiplayer interactive storytelling systems work. Similar to what happens with a single player interactive system, the problem of exception, that is, of something unexpected due to user interaction that was not predicted, was greatly increased. It uses a partially ordered planning algorithm to control the story generation and adaptation, with the addition of a “repairing” system to remove from the plan (the plot) elements that no longer makes sense after user interaction. This system differs from Logtell mainly because spectators play different roles, as players, while in Logtell the objective is to watch a story, being able to influence other characters, without the compromise of following and controlling only one of them.

One of the most similar projects to the model presented here is the Shapeshifting Media [13], which creates interactive screen-media narratives. It was broadcast on Finnish TV and provided an interactive experience that allowed spectators to participate by SMS messages, internet, and television. One of the main differences with this thesis is that it is centered on narrative objects, pre-recorded scenes, each defining their interaction points; that is: Shapeshifting Media system uses branching techniques, instead of a fully-fledged logical system, to create stories focused on their plot events definitions. In other words, it always uses fully instanced and pre-recorded events, instead of allowing more flexible plots to be constructed based on the logical structure of the story.

The approach presented by [14] uses StoryML, an XML based specification for multiplatform interactive stories, to build the TOONS application, featuring a story involving several devices: a TV screen, a light, an audio device, and a toy robot. It supports multiple environment presentation and interaction, what makes it similar to the architecture presented in the present thesis. However, there are a couple of important differences from the presented system.

One of the key differences with this thesis is that TOONS is based on presenting distributed media objects in multiple devices, where objects might even be something rather abstract like emotions. The approach of this thesis is, however, to present a single story on multiple devices with different resolutions and interaction interfaces, through a continuous video stream. Also, TOONS has no real logical engine. Another difference is that the media objects must all be created beforehand, while in the presented system the plot can evolve into multiple and surprising storylines.

Other works with multiuser interactive storytelling systems focus on local multi-player environment, as in [15], where users interact by drawing objects that are then transferred into the story in a virtual reality environment. In [16], multiple users interact with characters in different ways, using Wii videogame controllers, mobile phones, among others. However, those works differ from our work because they are focused on local interactivity instead of networked multiplayer systems.

There are already several different storytelling research works that approach partially what would be needed for a massive stream based interactive storytelling. On a general view, for an individual experience, Façade is the closest thing to the state of art in terms of interactive storytelling. However, it is based on a single user experience, and only available for computers.

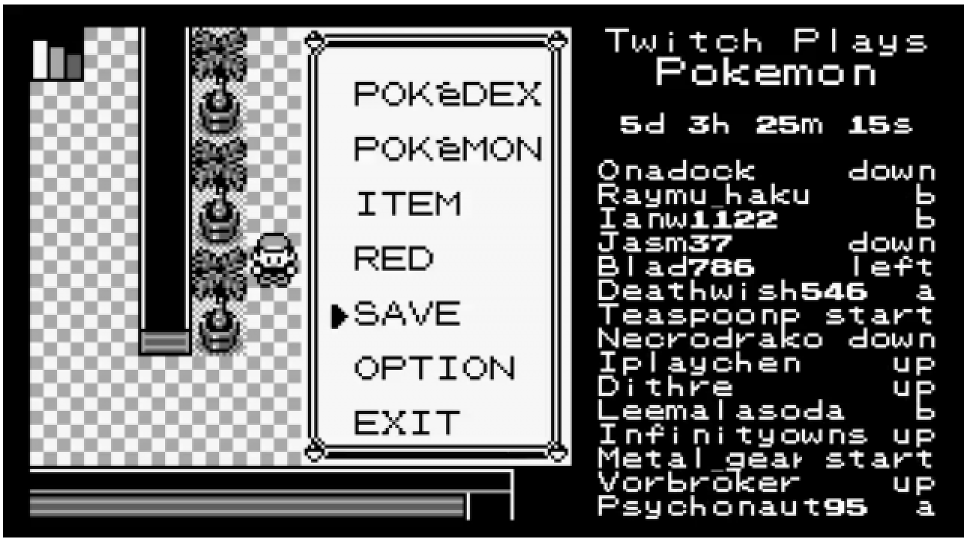

Figure 5 - Twitch plays Pokemon [34]

Although not really a traditional interactive storytelling experience, a recent streaming channel has shown how a mass of users can interact together with the same content. [34], in Figure 5, relates how a user created a system in which a videogame of *Pokemon* could be played through the website's chat functionality, by creating a "bot" that parses game controls and forwards then to a videogame emulator. The game was played simultaneously by more than 80 thousand players in peak hours. Even though sometimes generating chaotic results, players were able to surpass many of the game’s challenges, which was only possible due to the game design itself, since controls are turn based and do not have to be entered in real time.

Eventually, a simple voting system was created to pick whether the users wanted anarchy (any control is accepted) or democracy (the chosen commands are voted and then the most voted one is accepted after a number of votes). When in anarchy, controls are executed faster, especially since the stream itself has already a delay, but game progress is more exposed to be hindered by users that want to cause intentional chaos, causing intentional useless backtracking. In democracy, game can advance further in less time, but it can cause frustration due to the time it takes for the player to do something while voting happens, and also a part of the users are watching specially due to the struggling to win the chaos and the sense of winning together against the confusion.

Also, another interesting part is that users were so entertained by the experience that they began to create art and images inspired by the many situations that occurred during game play, like trying to use repeatedly an item that would not cause any effect due to being used at an unexpected time. And thus, a completely unexpected "story" arises not only from the normal gameplay, but due to the multiplayer experience.

However, even being a massive experience, it uses a normal videogame, which was not in any way prepared for this type of input in its gameplay. Most of the time is spent with almost random behavior, which the player moving around in erratic patterns. The popularity of the experience can be attributed not only to the novelty and the mass factor, but to the popularity of the game itself.

2.2. Logtell

Among available interactive storytelling systems, Logtell is the one that is most comparable with the proposed model. Logtell is a interactive storytelling system being developed for some time. Its approach seems to ideal as it has its focus on keeping coherent stories while allowing the user to interact. The user may not interfere in the story, or have a stronger participation, for example, establishing that certain events or situations occur, as long as they are logically coherent with specified model for the used story genre. One of the main characteristics of Logtell that is not ideal for the proposed model is that dramatization occurs on the clients, for which there are some ideas shown in chapter 4.

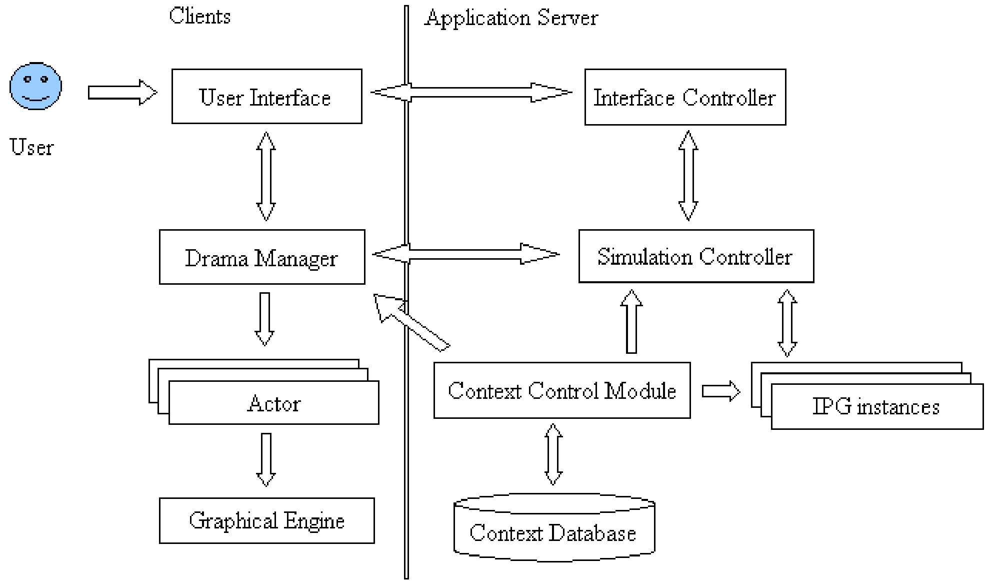

Figure 6 Current Logtell Architecture

2.2.1. Story Generation

The approach used in Logtell tries to generate and dramatize varied stories for a different genre using as base a specification through formal logics. The main idea is to allow that a user can interact as long as the coherence is kept.

Logtell, by keeping logical coherence in its story generation strategy, has a mainly plot-based aspect, although conciliated with character-based characteristics. It is inspired mainly in Propp ideas [2], extending, though, its informal notion of functions adopted on its research. Typical events are described through parameterized operations with pre-conditions and post-conditions, in a way to be applicable by planning algorithms. The character-based aspect is defined by goal inference rules that provide the goals to be achieved the characters when some situations are observed.

Propp's functions are the base for plot generation of many interactive storytelling works, like mentioned before. In Propp's work, a large set of fairy tales plots were analyzed and from them were extracted the functions that correspond to typical events that tend to combine according certain patterns. In a similar way, other literary genre can be analyzed so to extract the typical events and combination patterns.

In conventional stories, the events that occur usually do not happen in a chaotic and random way. In most of conventional stories, there is always some kind of logic behind the events, in general tacit, but understandable. Spectators or readers already accustomed to stories of a certain genre can easily determine if a given story is according to the implicit logical patterns on the genre. Events happens so that objectives (of the characters or of the history as whole) are attained (or frustrated) or to produce the conditions that allow the occurrence of other events. The events themselves end up changing the world, which can cause that other new goals occur, so that the story develops. This way, if a model for automatic generation of plots wants to guarantee the coherence of narratives, this model must, somehow, capture the "logic" imbued in the genre that it is using.

The context of stories in Logtell contains the following information about the plots to be generated:

- a set of parameterized operations with pre-conditions and post-conditions, specifying logically which events can occur

- a set of goal inference rules specified on temporal modal logic, declaring which situations can lead characters to pursue goals

- a set of facts that define the initial configuration of stories to be generated

- a set of suggestion inference rules, that analyze the story and provide a list of suggestions of interactions to be done by the users

The interactive storytelling experience in Logtell has a clearly different paradigm than games, reading a book or watching a movie. In games, the user interacts constantly, but usually the content of the story itself has a secondary importance. In the experience of watching a movie or reading a book, the content of story is fundamental, but there is no interaction. In Logtell, the user is also a bit of an author, since as the possible event sequences are not fully pre-established, and may be influenced decisively by user intervention. So that interesting stories arise dynamically, however, an authorial effort is necessary to define this good set of rules and events.

Plots in Logtell (in the main version of the system) are generated by a Prolog module called IPG (Interactive Plot Generator) [2]. IPG generates plots in multiple stages that alternate between goal inference, planning and user interaction.

IPG contains two sub-modules: one for goal inference and one for planning. Its planner is an extension of Abtweak [26], which is an hierarchical non linear planner. The planner allows the definition of an hierarchy of pre-conditions, in a ways that there is a prioritization in the search of solutions. On the other hand, since its a non linear planner (a planner that plans on the plan-space) works with partial event ordering. In planning in plan-space, it is adopted a least commitment strategy, according to which the order relations between the events and restrictions upon values of variables are established only when strictly necessary. The adopted approach helps, in special, the conciliation of multiple objectives, the generation of alternatives and the flexibility for the dramatization of plots. The IPG planner extended Abtweak with incorporation of Constraint Logic Programming [25] which helps the treatment of pre-conditions involving numerical expressions and concepts that allow the abandonment of objectives.

Story generation begins with the inference of goals of the characters of the story from the initial configuration. The system then uses the planner that, respecting the pre-conditions and post-conditions, inserts events in the plot so to allow reaching the goals. When the planner notices that all the goals were reached or abandoned, the first stage of the process ends. As the story generation resumes, if new goals are inferred, the planner is utilized again and thus successively. The process continues until no other goals are inferred.

Initial configurations and other story states are composed by sets of facts that describe the situation of characters and places and their relationships. In Prolog, an example fact would be that *Brian's current place is the Gray Castle* would be expressed by the predicate clause *current_place('Brian', 'Gray_Castle')* where *Brian* is a Character and *Gray Castle* is a place (used in the tested context used by the research). The facts in the initial configuration may be modified by the generated events (which are the executions of the planner operations) as the story unfolds. In Table 1 there is a list of the Prolog predicates used to represent situations in the used example context of fairy tales in Logtell.

The events inserted in story are instances of the operations defined in the context. An operation is an event typical of the genre (containing variables). Pre-conditions and post -conditions are specified using the predicates of descriptions of situations of the context.

Predicate	Description
knight	Defines a character as a knight
princess	Defines a character as a princess
magician	Defines a character as a magician
dragon	Defines a character as a dragon
nature	Defines a character as good, bad or neutral
strength	Defines a character's strength
alive	Defines if character is alive
place	Defines existence of a place
protection	Defines a place's protection level
home	Defines the home of a character
current_place	Defines where a character is
affection	Defines a character affection for another
hero	Defines a character as hero
victim	Defines a character as a victim
villain	Defines a character a villain
donor	Defines a character as a donor (of strength)

Table 1 Logtell Predicates

In Table 2 there is a list of the possible events that can happen in the example context.

Event	Description
go	Indicates that a character goes to a place
reduce_protection	Indicate that a character reduces protection of a place (ex: dismisses guards)
kidnap	Indicates that a character kidnaps another one
attack	Indicates that a character attacks the defenses of a place
fight	Indicates that a character fights another
kill	Indicates that a character kills another
free	Indicates that a character frees another
marry	Indicates that a character marries another
donate	Indicates that a character donates more strength to another
bewitch	Indicates that a character bewitches another (changing their nature)

Table 2 Logtell Events

The goal inference rules used by IPG are specified in modal logic formalism [24] that defines for characters in different classes (hero, villain, victim) which goals will be pursued when certain situations are observed. These rules use meta-predicates to talk about the occurrence of an event at a given time or about the truthiness of a fact (or its negation) in a given time. The formalism incorporates the treatment of partial orders of events, since the truth of a fact may depend on the order of the events.

A simple goal inference rule of the example, is that "*if a victim becomes unprotected and there is a villain around, this villain will want to kidnap the victim*". Another rule, almost complimentary, is that "*if a victim is kidnapped, and there is a hero that likes her, he will want to rescue her*". It is important to notice that the rules do not dictate directly the reactions of the characters. The rules only indicate goals to be pursued. The events that will be able to fullfill the objectives are filled in by the planning algorithm.

Table 3 shows the goal inference rules of the example context.

Rule	**Description**
The strongest hero wants to become stronger than the villain	When there is a hero and a villain in a given time, and the strength of the villain for greater than that of the hero, the hero will try to get stronger
Victim spontaneously reduces the protection at her current location	A victim, when in a place of the same nature as his/her own (good, bad), for some foolish reason ends up being unprotected, that is, puts himself/herself in danger
If victim's protection is reduced, villain will want to kidnap her	If a victim is in a sufficiently unprotected place, a villain will try to kidnap him/her
If victim is kidnapped, hero will want to rescue her	If a victim is kidnapped, a her will try free her
If victim is killed, hero will want to avenge her	If a victim is killed, the hero will try to revenge her
If the affection between two persons is high they will want to get married	If the affection between 2 characters is big enough, they will want to get married

Table 3 Logtell Goal Inference Rules

There are also the suggestion inference rules, which are similar to the goal inference rules, but made so that the user can suggest some event that is promising to move the story further, triggering goal inference rules. An explanation of some example rules used in the testing context follows in Table 4.

Suggestion Rule	Description
Causing a Villain attack	If there is villain alive and there is a victim in a place of different nature, the user may suggest: the villain to attack the victim place; the protection of the victim's place somehow be reduced; or the victim to wander to a unprotected place
Heroes fight	If there 2 heroes alive and both like an alive victim, the user may suggest them to fight
Causing Villainy	If there is a victim in a unprotected place, the user may suggest: the villain to kidnap the victim; the villain to kill the victim
Heroes destiny	If there is a donor and another hero around, the user may suggest: to donate power to the other hero; or to have the villain kill the 'main' hero
Saving the victim	If there is a hero strong enough around and a kidnapped victim, the use may suggest: the hero to kill the villain; or the hero to free the victim
Revenge	If the villain kills the victim the user may suggest the hero get more strength; or the villain to kill the hero
Love for the hero	If a victim likes a hero and both are not married then the user may suggest both to marry
Jealousy	If two characters like the same victim enough (and the victim likes one of them back enough), the use may suggest one character to kill the other one

Table 4 Logtell Suggestion Inference Rules

Logtell's interactivity is focused on giving the user means to explore the story's alternatives, being coherent with the logically specified context. To guarantee this coherence, interaction is always indirect. The user does not intervene in the story as a character or directly manipulates objects and positions.

Since its original version the Logtell system allows the user to interact using a step by step plot manager (Figure 7), controlling the IPG to generate stories and a Drama manager to render events. It was then changed to a client-server system, but keeping this functionality, which, in the end is used by the continuous mode. Users can generate parts of the story who can then interact and render the partial story, or continue the story generation.

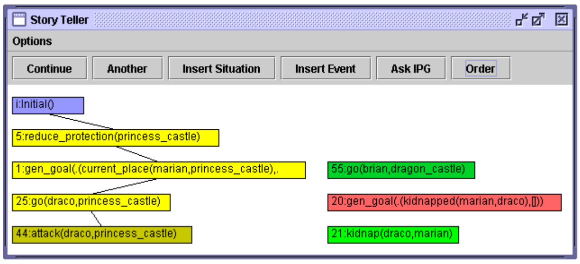

Figure 7 Logtell Step-by-step Mode

2.2.2. Step-by-step mode

In the step by step mode (Figure 7), users can see the events and objectives generated for the plot. By using nodes in a graph, the temporal restrictions are shown. To ease the search for goals, IPG works with partial event ordering, where it is only established that an event must precede the other when necessary. So that dramatization is possible, however, it is needed that the events are completely ordered. This order must be defined by the user in this mode. Then the partial story or the full story can be rendered.

For "weak" interaction, there are two main controls: *another* and *continue*. The *another* control requests a rollback and an alternative for the last simulation phase done, that is still unconfirmed. The *continue* control confirms the partial plot and also can be used to continue the step-by-step plot generation, with goal inference and planning phases.

The "strong" interactions can be done either by the *insert situation* command, allowing the insertion of goals to be reached, which are then pursued by IPG. Note that the inserted situation may fail in case there is no possible plan or of the computational effort exceeds configured limits. Another way to do strong interaction is by using the *insert event* command. As in the previous case the *continue* control must be used to validate and use IPG to achieve them.

Even though this step-by-step planning mode is not really used in the presented model, it is important to understand since it is in fact, the base of how Logtell's planning process occurs. Also this form of interaction is important to help authorial efforts when validating the designed contexts, similar to debugging programs.

Logtell is a interactive storytelling system that strives to be compatible with the concept of interactive digital television [5][6]. As such this model supports:

- allowing the generation of multiple stories, coherent and accepting user interaction;

- allowing continuous flow, where the story generation happens in parallel with dramatization and interaction;

- have a scalable architecture;

- presenting new ways to interact, adequate to different spectator modes and presentation ways;

As its model evolved, it uses simulation based on logic for story generation, as a way to help attending the coherence needs. On the other hand, it allows multiple interactions to be done on the story, whenever the user wants to. In its current state, the system focuses on keeping the story flowing and simplified interactions occurring.

2.2.3. Continuous Flow

As a way to give the final user, who watches the interactive TV, a pleasant experience, Logtell has the objective to show its content in a continuous way, without interruptions. As such, it uses coordination techniques as to generate the story in parts and, while a part of the plot is dramatized, new parts are produced. In its previous versions, plot used to be generated separately from dramatization, where the user could explore different possible stories by trying the story generation and its alternatives, and only then opt to see the story rendered. This was changed because it would require the user to stop step by step and interact, which is different than the usual television experience where story flows.

With its continuous flow of generating and dramatizing the story, Logtell allows to potentially more interesting experiences for the final user of interactive TV, with its continuous interactive paradigm. The user is in fact watching a story, and according to his/her desires, modify it, if so is desired. The user can be a passive spectator most of the time, if that is what is wanted, thus following the concept of a lazy interactivity. On the other hand, strong interactions can be made with little effort, taking the plot to different paths. Besides, some users may want a more active stance and interact more often.

As a way to support the continuous flow, Logtell uses the concept of chapters. This concept, already present in narrative structure, is consistent with the system's plot generation engine (IPG), since a generation phase is composed by a generation step followed by the inference of planning goals and plan generation. A chapter in this continuous flow correspond to a generation phase followed an automatic event sorting, since there would be no user to manually sort these events (this would be possible in previous Logtell versions, as long as event dependencies were respected). Since the story is dramatized in parallel, it is assumed that any interaction can only change future chapters, since it does not make sense to change part of the story that was already watched. A chapter then specifies a part of the story where one or more goals were inferred or resulted from user interaction, and by planning, events were inserted to reach those goals, who received a complete automatic order, compatible with the partial order produced by the planner system.

One of Logtell's main concerns is keeping a continuous flow while the story is generated and dramatized. To reach that goal, it has an distributed client server architecture that renders the story in a rich 3D client, while the server is a scalable architecture that can generate the plot in parallel.

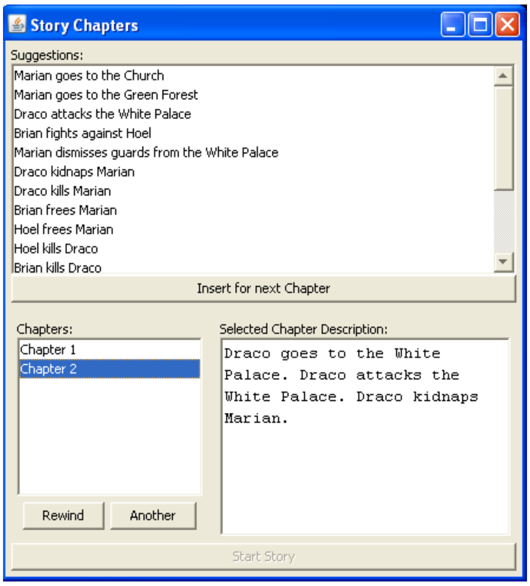

Figure 8 Logtell's continuous interaction

In its continuous interaction mode, shown in Figure 8, rich clients access the server using EJB calls. Events and situations, described in natural language, are suggested through a list that is continuously updated and sensible to toe the context that is being dramatized. Suggestions can be picked by the user so that Logtell can insert them in the next chapter. Chapters are also listed and there is a window where the events of a selected chapter can be seen in summed up way.

There is also the Rewind and Another weak interactions that allow the generation and dramatization processes to go back to a past chapter.

Logtell's model aims toward attending to multiple users, for that it has a simple voting mechanism. In the multiuser mode, the same rich clients connect to the story generation servers (that write the plot), while each client renders individually the story, according to the story chapters being watched. Each user can vote on desired suggestions, and the most voted suggestion is chosen.

2.2.4. Dramatization

A important part of Interactive Storytelling is Dramatization. Not only there must a be plot, but stories need to be represented somehow, and so, this can be done in multiple ways, may them be graphically or in some other way.

Logtell's Dramatization (Figure 9) uses its own graphical engine, implemented in Unity [19]. This is a complex graphical engine that uses GPU acceleration and allows (with 3D rendered graphics) rich clients to render the story that is being generated, based on its events. Each event is rendered, in real time, synchronizing the characters actions and controlling how they interact with the scenery.

Figure 9 Logtell 3D dramatization

This complex Drama manager needs to be tied with the story contexts definitions. This means that even though it is a beautiful rendering engine, for each different story, the different clients will need to be updated to show the different stories and their 3D models, backgrounds, and other resources.

Even if Logtell's model is able to attend to multiple users, still it has some limitations that should be surpassed in order to have a true massive system. As of its current version, it needs rich clients to be fully re-implemented whenever a new platform is added. This means that for each platform, their respective graphical APIs must be used, in order to be able to render the stories. This also mean that the capacity to render stories may be limited by the clients' graphical capabilities.

Also, this means that to use the current implementation, server calls in Java or Webservices must be implemented, which demands more effort and may be prohibitive in some platforms.

The multiuser aspect of Logtell is interesting, but still not completely adequate, since it has no actual strategies of how to handle that a mass of users may vote in multiple choices, and only the most voted one is picked, as long as it is logically sound. This means that potentially this could cause frustration, since all users would be always treated the same, plus being limited to only seeing one suggestion unfold in the story each chapter cycle.

2.3. Digital Television

In recent years, the concept of Television is changing. Analog Television is being replaced by Digital Television, which presents superior image and sound qualities, as well as more interaction possibilities. Besides, new forms of media are appearing, and their widespread use is everywhere, like broadband internet, and Smartphones/Tablets with 3G and 4G connections, which are interesting alternatives for watching content that would be otherwise shown in TV in different and fixed times. These changes result from the evolution of digital technology and digital convergence, which brings together multiple improvements and innovations in multiple things, such as in the network communication infrastructures, in software and hardware for compression and data transmission, and in broadcasting services [41]. As a result, multiple technologies arise in order to change the way people watch and interact with TV. These changes regard not only image and sound quality [42] but are also related to the available content and its relation with the spectator.

In this new scenario, one important question is about the amount of interactivity, which brings to the development of Interactive TV [43], or simply *iTV*, as an important area for both academy and industry, due to its potential to reach lots of people.

In the theories of media and mass communication, some researchers [44][45] investigate the gratifications and uses of media, and what are they used for. This is an approach known as the “Uses and Gratifications theory (UGT)”. In [46], the forms of gratifications are separated in four different categories of human interactions: personal relationships, personal identity, surveillance, and diversion. These categories can be understood as four basic needs that a user has. Analyzing these forms of uses is important to better understand how interactive television can work.

Personal Relationships: people use the media to fulfil their need for companionship. This is a form of use that brings together people somehow. These uses (*i.e.* needs) are characterized by Social Interaction, Communication Facilitation, Affiliation and Dominance/Competence. Basically it focuses on creating the sense of community.

The research in [47] shows that the shared experience of television works as a social aggregator. This shows the importance of the shared experience that interactive TV can promote. This sense of belonging is characterized by two main categories [44]: "*Firstly, viewers are able to place themselves in a specific social and economical context, either by comparison with different groups or by identification with their own. Secondly, viewers are able to discuss with other viewers what they watched on television, and thus be able place themselves in a community of viewers and to interact socially with others*".

Diversion uses are the main job of television, including Relaxation, Escape, Arousal, Passing Time, Habit, Escape, and Entertainment. This category focuses on entertainment and avoiding daily routine. Some examples are comedy, sports, competitive games, music, and even the thrill of following an election [45].

Surveillance encompasses the uses focused on learning about the world, like news and keeping current with the events that are in tune with the viewer's interests. This includes uses for Education, Information, Guidance, and Learning. The main objective is to acquire knowledge and advice in general for education as well as help in decision making.

Personal identity is the kind of use in which individuals identify themselves in comparison to a broader culture. For instance, when a spectator imitates some character in a TV show, using similar clothes and phrases; when they begin to act in similar ways to characters they liked or felt identified with. These include uses for Cultural Satisfaction, Identity Formation and Confirmation, Lifestyle expression, Social Learning and Reinforcement of Values.

Traditional television has shown over time an increasing number of TV shows in which spectators are invited to participate somehow, such as reality shows with voting. This demonstrates the existence of a potential to explore over possibilities of interaction. The iTV would, then, be an evolution of the analogical media, aggregating interactivity to video and bringing the opportunity of immediate feedback from the spectator to the content providers, who sometimes are also the producers. Among a number of possibilities of interaction, we can emphasize the following:

- a weak form of interactivity, in which there would be the possibility to re-watch shows at the desired time, skipping ads, etc; acquiring more information about what is being shown, whether movies, news, sports, etc.;
- showing directed advertising and remarketing, with the possibility of sales;
- effective interference with the content being shown, changing for example, the ending of a story being watched;
- the massive interaction of a group of users watching a shared content;

The last two items are specially interesting for the development of applications targeted to entertainment and education, which are the focus of this work. More specifically speaking, the present work is focussed on the use of interactive storytelling within a perspective of interactive TV for multiple users.

2.3.1. Interactivity and Television

One of the main changes that comes with iTV is exactly the way Television is used. Now, the user is beyond a simple spectator in the simple meaning of the word, being able to interact with what is being watched.

It is important to noticed that interactivity with television shows already exists for some time, although in a more improvised and less dynamic way. One example is with television shows where the spectator can call to make decisions, for instance in reality shows. Also there are game shows where spectator can call and answer questions for prizes, which are even older. There were even games like "Hugo" in which spectators could play a videogame by using the telephone keys. In Brazil, a TV show called "*Você Decide*" ("*You decide*"), which is somewhat similar to some aspects of the present research, allowed spectators to choose an alternative for a story's protagonist to take. Usually those choices were associated with moral choices and polemical situations. The most voted situation was chosen and then shown, in the end. One interesting aspect is that it was broadcast without pause: the story presents the dilemma/choice to be done, and the story goes on, and while the spectator watches and is given more context to make a choice.

In any case, this demand for an "improvised interactivity" serves to show that there is a potential for shows in which spectators can interact, instead of only be passive spectators. With continuous technological advances, users become more skilled in interaction through Smartphones, tablets, videogames, computers, etc. Thus, this defines a scenario in which interactivity in TV arises as a natural evolution.

There is a variety of different approaches on how to incorporate interactivity mechanisms on TV. There are approaches that focus on a greater interactivity, while others promote less interaction effort. The capacity to interact usually depends on the computational power of the devices that receive this TV signal and process it, usually in set-top boxes. There is the so called lazy interactivity, in which the capacity to interact is more limited, demanding less powerful devices and not demanding too much attention from the user, which tends to be more natural for spectator, more used, when watching TV, to a more passive stance. On the other hand, there is the other approach that focuses on using a greater processing power on reception devices in order to have a more powerful interaction. Instead of a lazy interaction, the focus of those application are in more complete and demanding interfaces, which are then more similar to interfaces of PC applications, for instance.

Regardless of the approach for interactive TV, the term spectator tends to lose its meaning, since the user no longer will be an individual that only watches content sent to a TV. The user then, in the same way as when using a Computer, to be an active agent in the process, which shows how the term becomes inadequate. The difference between the approaches correspond to the level of activity that may demand more or less attention and that can lead to bigger or lesser changes on the content that is being watched.

Following a natural expected evolution, interactivity will probably grow with time, demanding the creation of new business models and infrastructure resources to support them. One of the bottlenecks for the development of more complex applications is the computational power of set-top boxes and other reception devices, in all their diversity. The communication network must be adequately prepared to respond and process in time the desires and demands of the users, since it will not be dealing with only one static content being broadcast equally and in an unidirectional way.

2.3.2.
Applications and Services in Interactive Television

Besides infrastructure issues, efforts are needed for the creation of appropriate content for the new media. The content generation for this new environment demands work in project and programming, besides involving the creation of new research areas and the integration of multidisciplinary knowledge, in order to adapt them for this new scenario.

Also commercial usages will have to adapt for the interactive television. The usage of advertising on open TV, for instance, will adapt itself to the new ways. New business models will surely have to be created and tested.

The applications for interactive TV can be of the most varied types. We present bellow some examples of the types of interactions that can happen on these applications.

Interaction for Content Selection

One of the main forms of iTV applications is in the form of programming guides that show, through graphical interfaces, varied information about the programming and helping guides to the user. These information allow the user to pick programs and schedules, to buy content through pay-per-view and other forms of Video On Demand and other miscellaneous content like radios, horoscopes and weather forecast. These guides are also known as Eletronic Program Guide (EPG).

Since this information is in a digital media, there is the possibility to easily record content, by using personal video recorders, also supported by some televisions and cable set-top boxes. Since metadata is available, users may be capable of searching for movies with specific criteria to their liking. Video on demand is a service hat allows the users to buy and watch content when desired.

In the recent years also many different Streaming services have appeared, with the most popular one being Netflix. Netflix is a subscription based service that allows users to watch video on demand, through streaming, supported on multiple devices like some internet enabled Televisions, Personal Computers, Smartphones, Tablets and Videogames.

More advanced interaction mechanisms will allow the creation of a more individualized television experience. Eventually in news it will be possible to watch personalized categories of news; interactive documentaries, working like digital encyclopedias, and other similar ways to access informative content on what is being watched can be implemented. Also it should be possible to individualize the way the same content is shown. Users should be able to watch sports through different camera angles and replaying favorite moments.

Internet Services and Interactive Portals

By having a return channel, comes the possibility of having services like internet for navigation and email, and also other applications like instant messaging, chatting and social networks.

As a form of interactive services, there are also portals, which usually can work as *walled gardens*, that can also be defined as a restricted number of web pages and/or content. Inside these portals can access interactive applications that can include games, news, apps in general.

Games

One of the applications of interactive TV is in the form of games. Games can be used to educate and entertain. Also they can be exploited in movies and products advertisement.

A possibility is to have games being played individually on set-top boxes, which usually is available in cable TV services, with simple games that mimic classic games. Another, more demanding way, is to use a return channel, so that users can interact with themselves, bringing the now common multiplayer experience from videogames and internet games to the TV set.

Elements mixed with TV Content

Another way for interactive TV is to use elements mixed with the content being displayed, that is, interactive elements that appear while movies and TV programs are shown, in parallel or over the presented video. To use these elements usually semi-transparent graphics should be used, appearing in specific moments in the television borders. It is possible, for example, while a program is shown, to allow the user to buy a product that appears or allowing the user to order, for example, the same food that a character is eating.

This type of interactive TV use can be applied for advertising and TV commerce, also known as t-commerce. Besides, the graphical elements can also be used to show statistics, or any other type of information, like statistics, in the case of sport events, or even as an educational support for a movie, for instance, explaining details and providing the users a support tool for the content shown.

Interaction with the Programming Content

The most revolutionary perspective for interactive TV is the possibility to interact with the content that is being watched. This perspective, however, is the most challenging one, and connected to the work presented here. There must be means that, for example, the story can unfold normally while the spectator watches and interact with the story.

To implement advanced interaction mechanisms like these, there are some difficulties regarding the systems' architectures and business models. There is a general lack of world accepted standards and a feedback or return channel is needed. On the other hands, there are some environments where TV is free and financed by advertising mostly, the business model may need to adapted, since there are a lot more of content options and the content is much more controllable by the user.

2.4.
Voting Systems

In this section, we go over the ways voting methods function. An important part of this work is to propose a model to handle users' votes. Also this section shows that voting methods are not absolute and there is no perfect “democracy” of votes. Voting systems are taken into consideration in the model presented in this thesis.

Whenever there is a group of people and there is a decision to be made, there are many ways to proceed. If the intention is to have a collective choice, in some way, more than one person can manifest himself/herself. In other words, it is needed some way of combining individual opinions to reflect what “the will of the group” is.

One of these decision making processes is voting, where each of the individuals somehow will have his/her opinion represented, and then something is decided. This section will go through how voting is usually handled, and what are the usual ways of how to reach group decisions.

A voting system is a method that allows the voters to choose between options. It defines a procedure since it should enforce rules to ensure that the voting is valid, and this includes many aspects: how votes are expressed, counted and aggregated in order to reach a result.

Usually if a voting is more informal, there may not be strict rules for decision making. But in a voting system, usually there is a decision rule, quota and a vote proposal. The most common voting systems are **plurality voting** (with a number of variations, such as preferential voting, first-past-the-post voting, voting with elimination, etc.), **majority rule**, and **proportional representation**, which are usually used for voting on a motion[1]. A more comprehensive overview on the different voting methods can be found in [28].

2.4.1. Basic definitions

A voting body can be formally presented as a set *N*, containing every member in the voting body. The *W* set includes all possible winning subsets (called the *winning coalitions*) , that is, all subsets that can accomplish their goal in a voting situation. Some definitions can help us to understand voting systems, as follows:

- *weight*: the weight w_i of each voter *i* is the number of votes he/she controls (*e.g.* a board of directors may have "director-1" with 4 votes, as the largest stockholder of the company, "director-2" has 3 votes, and "director-3" has 2 votes: $w_1 = 4$, $w_2 = 3$, and $w_3 = 2$).
- *quota*: the quota *q* is the minimum number of votes needed to pass a motion.
- *weighted voting system*: it is a system in which the preferences of some voters carry more weight than the preferences of other voters. We denote this type of system as $\{q: w_1, w_2, \ldots, w_n\}$.
- *dictator*: a dictator is a voter who has enough votes to pass any motion in a single hand (a dictator's weight w_d is always greater than or equal to the quota: $w_d \geq q$).
- *veto voter*: a voter that is not a dictator but can single-handedly prevent any group of voters from passing a motion is said to have a veto power.
- *dummy*: a dummy is a voter with no power.

[1] A motion is a formal proposal (for action) made by a member of a deliberative assembly. Usually a motion represents a voting procedure involving only two alternatives.

- *coalition*: a coalition is any group of players that join forces to vote together (in order to accomplish their common goals). With *n* voters there are 2^n-1 possible coalitions.
- *wining coalition*: it is any group with enough votes to win.
- *losing coalition*: it is a group without enough votes to win.
- *minimal winning coalition*: a group in which the desertion of a single voter turns the group into a losing coalition. We denote this desertion operation of a voter *i* as $S\backslash\{i\}$ or simply $S\backslash i$. The analysis of minimal winning coalitions reveals voting strategies [2].
- *critical voter*: it is a voter whose desertion from a winning coalition turns it into a losing one.

Some restrictions can be applied to *W* [27]:

(i) $\varnothing \notin W$, that is, there is one voter at least;

(ii) $N \in W$, that is, the set of all voters is a winning coalition (called the *grand coalition*);

(iii) *Monotonicity assumption*: $S \in W$ and $T \supset S$, then $T \in W$, that is, if *S* is a member of the set of winning coalitions *W*, then any set that contains *S* is also in *W*. The motivation is that having more voters in a group that can already accomplish its goal can do no harm in terms of "winning".

The above-mentioned definition of a voting system is aligned with the definition of a simple game in the sense of the game theory. The notion of a simple game was introduced by [29]. A simple game is an *n*-person game defined as a pair *(N, W)* satisfying conditions (*i*), (*ii*), and (*iii*) mentioned above. In a simple game, we also define the characteristic function of a coalition *S* as $v(S)$ such that $v(S) = 1$ if *S* is winning and $v(S) = 0$ otherwise. In a weighted voting

[2] The weight of a voter is not a good measure of a voter's power. Suppose that $w_1 = 4$, $w_2 = 3$, and $w_3 = 2$, and $q = 5$, *i.e.* {5:4,3,2}. In this case, the minimal winning coalitions are: {1,2} and {2,3}. That is: none of the players can pass a motion alone (they have the same power, although they have different weights).

system, $v(S) = 1$ if $\sum_{i \in S} w_i \geq q$, otherwise $v(S) = 0$. We should notice that a voter i is a critical voter in S iff $v(S) = 1$ and $v(S \backslash i) = 0$.

Indices are used to measure how important a given voter is among the group of critical voters that can bring down winning coalitions. They are called *power indices*. The most widely used index is the normalized Banzhaf power index, which is defined as:

$$\gamma_i^B = \frac{\#\ of\ times\ voter\ i\ is\ critical}{total\ \#\ of\ times\ all\ voters\ (together)\ are\ critical}$$

that is:

$$\gamma_i^B = \frac{\sum_{S \subseteq N} [v(S) - v(S \backslash i)]}{\sum_{i=1}^{n} \sum_{S \subseteq N} [v(S) - v(S \backslash i)]}$$

The problem of calculating Banzhaf indices of players is #P-complete. Even if we only want to calculate probabilistic Banzhaf index of the biggest player the problem is NP-hard [35].

A discussion about game theory and power indices can be found elsewhere [38]. Some more recent works propose the representation of power as the frequency voters build coalition together according to the structure of the game [39].

A voting system also defines a form of ballot that is the set of possible votes. There must be also a tallying method, or algorithm that can determine the outcome. This outcome may have a single winner or multiple winners. The system may also specify how power is distributed among voters.

2.4.2. Criteria and Methods

It is very difficult to compare different voting systems because the voters can influence the systems in many ways, since it deals with opinions. One way to compare them is by using Criteria. Mathematical criteria can always be used to make a choice either pass or fail, which gives objective results. Even so, it can be debated whether this is “correct” or not.

There are some fairness criteria that help defining potentially wanted properties of systems mathematically. Some of those criteria are:

[1] **Majority Criterion**: If there is a candidate that is the first choice of a majority of the voters (more than half), then this candidate should be the winner.

[2] **Condorcet Criterion**: It there is a candidate that wins on a one-on-one comparison between itself and any other choice, then that candidate should be the winner.

[3] **Monotonicity Criterion**: If a given choice C is the winner of an election, and in a reelection, all voters who change their preferences do so in a way favorable to C (and ONLY C), then C should be the winner.

[4] **Independence of Irrelevant Alternatives Criterion**: If C is the winner of an election, and one or more of the other choices are removed and the ballots are recounted, then C should be the winner of the election.

There is also **Arrow's Impossibility Theorem** [30], that dictates that it is impossible to devise and election method that satisfies all these four fairness criteria. It also says that the sum of the individual rationalities does not produce a collective rationality. This is important to consider, that despite efforts, voting is not "perfect" in a sense of building a full rational group decision. Thus we can conclude that making choices in a consistently fair way is inherently impossible in a democracy.

Single winner methods vary on their ballot type. In single vote systems, a voter picks one choice. In ranked systems, each voter ranks each candidate. There is also rated voting system, where a score is given to each possible candidate.

The most common single-winner voting method is **plurality**, where each voter can choose one candidate, and the choice that gets the most votes wins, even if not by majority. It satisfies the **Majority** and **Monotonicity** criteria, but violates the Condorcet criterion.

The principle of **Majority rule** defines that the winner candidate has more than half the votes. This is much harder to accomplish if there are more than 2 candidates. Sometimes in order to achieve this majority using plurality, multiple rounds are done, eliminating one or more candidates.

In **ranked voting** methods, also known as **preferential voting** methods, each voter ranks the candidates in order of choice. Sometime it is not needed to rank all candidates, in which case unranked ones are considered as tied for last place. Depending on the method, voters can give the same rank to multiple candidates. A interesting reference that can be applied to the present thesis can be found in [31].

A common form of **ranked voting** is **instant runoff voting**. This is useful to get voters preference all at once instead of multiple rounds of counting. As votes are analyzed, the option with fewest first choice votes is eliminated. In successive rounds of counting, the next preferred choice still available from each eliminated ballot is transferred to candidates not eliminated. The least preferred choice is eliminated in each round of counting, until a majority winner is reached. This method is also called **Plurality with elimination**. It satisfies the Majority criterion and violates both the Condorcet and Monotonicity criteria.

A form of ranked voting is the **Borda Count** method. In this method points are assigned for the position each candidate finishes on each ballot. So, 0 points for last place, 1 for second-to-last place, 2 for third-to-last and so on. The choice that receives most Borda Points is the winner. This vote method satisfies the Monotonicity criterion and violates the Condorcet and Majority criteria.

The **pairwise Comparison** voting method is one of the few that satisfies the Condorcet criterion. It works by using a preference (ranked) ballot. When comparing each choice to each other; whichever of the choices was more preferred by voters wins one point. In case of a tie, both candidates get half a point. Then all points are added, and the choice with the highest number of points is the winner. Although it satisfies also the majority and Monotonicity criteria, it can violate Independence of Irrelevant alternatives.

2.4.3. Weighted voting systems

We have already presented the concept of weighted voting systems in Section 2.4.1. In the present section we give more details and examples. A weighted voting system [32] is any voting method in which the voters are not necessarily equal in terms of their number of votes. Usually we present weighted voting systems as {q: w_1, w_2, …,w_n}. The quota q will always fall somewhere between simple majority and unanimity of votes. The quota is important because weight voted games are usually used for A/B motions. For example, q is needed to avoid **Anarchy** – *i.e.* when the number of both A and B votes are greater than the quota. Also **Gridlocks** should be avoided: when the quota is greater than the number of votes in the system.

Coalition is a set of some players of the voting system (if it includes all players, then it's a grand coalition). A coalition is formed to support or oppose a measure (that can consist even of only one player). A winning coalition consists of a set of voters with enough votes to pass a measure. A losing coalition does not have enough votes to pass a measure. A blocking coalition has enough votes to prevent a measure from passing.

Whenever a coalition needs a player's vote to be a winning coalition, then it is said that this player is a critical player. This means that if the player's weight is subtracted from the coalition's total weight, the remaining votes drop below the quota.

In coalitions there may be multiple or no critical players at all. For example, in the voting system *[6; 4, 3, 2]*, one of the winning coalitions is *{P1, P2}*, and its critical players are *{P1, P2}*; while another winning coalition is *{P1, P2, P3}* and in this case only *{P1}* is a critical player. Using another example system *[101: 99, 98, 3]*, in one of the winning coalitions *{P1, P2, P3}*, there is no critical player (because if remove one player at a time, the coalition still wins the quota). A minimal winning coalition occurs when each voter in the group is a critical player.

It should be noted that in Weighted Voting Systems, there may be some situations in which the weights can be deceiving. It is possible for a player with many votes have as much power as a player with few votes. These differences may characterize the players as Dictators or Dummies.

A dictator is a voter who can pass a measure even when all others oppose the measure. For example in the system *[10: 11, 6, 3]* even if P2 and P3 vote together, they can't win over P1 vote.

A voter has veto power when this support is needed to pass a measure. For example, in this system *[7: 4, 2, 1]* the only way to go over the quota is by having P1 choose it.

Dummy is a voter who is never needed to win or block a measure. For example, in the system {*9: 5, 5, 4, 2*} motions will pass if at least two of the first three players vote for it. Player P4 is always irrelevant, since it will only join a coalition that would already win.

In [33], it is argued that weighted voting may not be fair, due to these differences in voting weights and the facts that may be Dummies or Dictators. Since voters' powers are not so simple to perceive by looking at their votes' weights, this research defend using methods to calculate their actual voting power indexes.

2.4.4.
Some final remarks on voting systems

Voting systems can be compared: Two voting systems are equivalent if there is a way for all of the voters of the first system to exchange places with the voters of the second system and preserve all winning coalitions. Ex: *[50: 49, 1]* and *[4: 3, 3]* - unanimous support. Also, every 2-voter system is equivalent to a system with a dictator or one that needs consensus.

The work of [36] has an interesting insight for handling votes, of which there was an application in Nurmi's work [37]. It is suggested that to estimate a voter's capability to decide by observing past decisions that are aligned with the majority. In other words, the majority decision is considered as a plausible proxy for the "trust". What is interesting about is that it highlights the idea that voters' decision skills can be estimated by using a history of their decisions, even without domain specific knowledge of what is the "best choice" / "truth" regarding voting decisions.

As it can be noted, it is a complex problem of how to allocate voting weights. Another work that is worth mentioning is the Penrose Method [40], that was created for voting weight allocation of delegations in decision making groups, proportional to the square root of the total count of the population in the delegation. This is based on the square root law of Penrose, that a priori voting power (using a Banzhaf index) of any member of a voting body (the scenario used of multiple groups in which each group has a weight for their votes, for instance in the United Nations) is inversely proportional to the square root of the group size. It supports the idea that otherwise using proportional allocation would then result in excessive power for the large groups. Mathematically speaking, this method is interesting for the presented model because it can be used to remove excessive difference of voting weights.