Aspects of Interactive Storytelling Systems

David Shaw

Thesis submitted for the degree of Master of Computer Science

Department of Computer Science and Software Engineering
University of Melbourne

October 30, 2004

Produced on acid-free paper

Abstract

Storytelling has been popular throughout the ages in a variety of different forms, and now computers offer the potential of interactive storytelling; stories that can adapt to the wishes of the audience. However, to present a dramatic experience to compete with existing storytelling forms, any interactive storytelling system requires the ability to maintain the integrity of the story while coping with the audience decision about story path. This dissertation proposes a method for a core component of such a system, a model for abstracting plot to suit an interactive storytelling system.

Interactive storytelling is a medium where the audience is an active participant, taking the role of the protagonist in the story. For the storytelling to be truly interactive, the audience must be given the opportunity to shape the path of the story through their actions. A linear pre-written story cannot provide this level of freedom, as the story acts as a constraint to the audience's ability to choose their own path. For interactive storytelling to be a possibility, an algorithmic method of writing stories is needed.

To work towards solving the problem of integrating story with interactivity an abstract model of plot is presented, which uses the metaphor of doors and keys to represent plot challenges. In this model, 'doors' represent an important milestone or choice that the audience can make that affects the path of the story. 'Locked doors' are doors that require an additional element or action, for example fetching an important item. 'Keys' are the elements or actions that 'unlock' the locked doors. A prototype of this model is shown to generate abstract story worlds that would suit an interactive environment, and an example is given showing how this model can be used to generate plots for interactive stories.

Acknowledgements

I would like to thank my supervisor, Dr. Nick Barnes, for his advice and counsel throughout the course of my study. I also thank Dr. Alan Blair, my co-supervisor before his departure to the University of New South Wales, for his contribution both while at Melbourne and after his move to Sydney. Finally, I would like to thank the continual support I received from my family, in particular my parents, Catherine and Peter. I am grateful for all your support.

Declaration

This is to certify that:

(i) the thesis comprises only my original work towards the Masters,

(ii) due acknowledgement has been made in the text to all other material used,

(iii) the thesis is approximately 30,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed,

David Shaw

October 30, 2004

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Ah, cruel Three! In such an hour
Beneath such dreamy weather,
To beg a tale of breath too weak
To stir the tiniest feather!
Yet what can one poor voice avail
Against three tongues together?

Lewis Carroll - Introduction to *Alice's Adventures in Wonderland*[5]

Stories have been a popular form of entertainment for millenia. People have been drawn to a well told story, from the words of Homer's *The Iliad* through to the modern computer generated films of today such as *Finding Nemo*[57]. It is therefore natural for stories to be a vital component in computer games, one of the newest media that has the potential for providing drama. While there will always be a market for 'twitch' action games (such as *Quake*[18]) and puzzle games (like *Tetris*[38]) for which stories are not mandatory[1], most games benefit from having a well crafted plot. At the most basic level, adding a story gives a sense of purpose to the action in the game, linking otherwise unrelated tasks together to form a meaningful whole (this can be applied to practically any game that is 'level-based', such as most RTS[2] and FPS[3] games.) In some types of games (particularly adventure and RPGs[4]) advancing the story is a vital component of its appeal. For many games, the possession of a good plot is (or at least should be) a high priority.

In order to craft a story for a game, the most common and intuitive approach taken is to apply the ideas and techniques developed for the established traditional forms of story-telling. For text based games, methods applicable for novels seem appropriate for application, but with the audio-visual mastery of computers today, the 'silver screen' is the standard source for inspiration. By using film as its guide, the story for a game can be written essentially the same way as a script for a screenplay.

There are a number of advantages to applying the existing techniques of film to games. One is that as the same creation methods are being applied, the same type and level of drama as that which graces the traditional media can be achieved (provided the talent of those involved in creating the story is equivalent). Another is that traditional scripts are reasonably easy to write, edit, understand and apply, and people are comfortable working with them.

[1] Although even with these types of games it can be considered that there is some benefit from a containing a form of structure similar to that found with stories, even just to merely regulate the frequency of the action.

[2] RTS: Real Time Strategy

[3] FPS: First Person Shooter

[4] RPG: Role Playing Game

However, there is a major disadvantage to using film as the approach for scripting stories in a game. Film is purely linear, their narratives will never change. This is necessary because it is a passive medium, to experience the story all that is done is to watch from beginning to end. The viewer is put in the role of an independent observer, detached from the events shown on the screen. The same can be said about novels and traditional theatre such as Shakespeare. No matter how deeply moving a version of *Hamlet* is, it is not the viewer but rather another person (Hamlet) who lives out the story. The power of the story comes from the rich description of the narrative and the characters.

Computer games have the same audio-visual abilities of film, but with a crucial additional feature - interactivity. Unlike film, input from the player is used to affect the output of the game. The player can tell their character when and where to move, and which way to jump. This level of interaction greatly adds to the immersive effect of the medium, where the player projects themselves into the character which they control. The player is no longer witnessing the exploits of a character in some game world, to some extent it is *them* who is having the adventure. The character may be considerably less developed than Hamlet (as are practically all game characters), but the immersive nature of the game is such that the empathy can be much higher (as any misfortune that occurs to the character directly affects the player). This immersion with the game world is a powerful effect that can immensely add to the appeal of a game.

Herein lies the problem involved with applying a story to a game. In film, the director has complete control over the path of the story and the actions of the characters. In games, some control must be transferred to the player (otherwise there would be no game). Most games give the player a fair amount of control of their character's actions for the 'gameplay' sections of the game[5], but little to no control over the direction of the story[6], in an attempt to strike a balance between story and gameplay.

While a viable technique for adding plot to games, this approach sacrifices immersion for story. The level of immersion in a game is proportional to the amount of control given to the player. For the player to be truly immersed in the game world, they need to be given the option to choose their actions (and for them to genuinely affect the story). They need the freedom to make their own path and roles. This cannot be done with the linear film script techniques, as this 'free' approach is inherently non-linear. This means that a dynamic non-linear story cannot rely on using a traditional pre-written script to ensure compelling drama. New techniques need to be developed.

To consider new techniques for interactive storytelling, it is helpful to compare the potential using the range of an established medium as a 'thought exercise'. This will aid in the discussion of both the potentials and the problems with developing an interactive system.

Existing linear storytelling methods, while the most popular, are not appropriate choices for this task, as they do not allow proper exploration of the interactivity. Thus the cinema, although favourite in comparing business models (and success[7]) or cinematographic techniques[28], is not a valid

[5]For example, the 'battles' in an RPG, or the 'missions' in an RTS.

[6]Story elements are usually advanced in 'cutscenes' (often pre-rendered CGI movies or video clips), where the player has no control over what the characters do.

[7]Such as the oft repeated (and often misinterpreted) statistic that video game industry

medium for this exercise. For a workable analogy, a different medium is needed. One such medium, proposed by Laurel as an appropriate analogy for interface design as well as interactive fantasy, is the theatre; actors portray characters aided by props, backdrops, lights etc. in order to provide a performance for an audience. The exact details of who is involved or how the performance is made are irrelevant; for an audience captivated with a good performance, what is happening, and the story presented, is all that matters[22, p.14–17]. As long as the performance is engrossing, it does not matter if the actors are real or actually computer generated entities controlled by artificially intelligent algorithms.

There is, of course, a crucial difference between what is the traditional method of theatre and an interactive medium (as traditional theatre isn't interactive). For non-interactive theatre (the vast majority), the actors and audience are separated; the former perform on the stage and the latter sit in seats as observers. For interactive theatre the distinction is blurred; the audience are in effect on stage performing with the actors and influencing the story. This leads to some problems, as noted by Laurel:

> What would it be really like if the audience marched up on the stage? They wouldn't know the script, for starters, and there would be a lot of awkward fumbling for context. Their clothes and skin would look funny under the lights. A state of panic would seize the actors as they attempted to improvise action that could incorporate the interlopers and still yield something that had any dramatic integrity. Or perhaps it would degenerate into

sales are higher than movie box-office takings.

> a free-for-all, as performances of avant-garde interactive plays in the 1960s often did.
>
> Brenda Laurel, *Computers as Theatre*[22, p.16–17]

The dilemma facing the director of this interactive theatre is how to incorporate the audience into the performance while still retaining enough cohesion to maintain the story. The important question is who has the power to shape the story: the actors, who are under the full control of the director, or the audience, who the director has no control over but are the patrons of the theatre. The balance of power determines the type of performance that will be given, and therefore the type of interactive experience. There are three different approaches to solving this problem, which result in three different styles of stories:

1. The actors retain full control over the story, giving little or none to the audience.
2. The audience is granted full control over the story, leaving little or none for the actors.
3. Both the actors and the audience share control over the story.

The vast majority of traditional storytelling media of film, literature and theatre gives the audience no control over the story. If the storytellers retain full story control, they have complete power to tell whatever story they wish, using whatever techniques they can apply. The complexity of dealing with the unpredictable nature of the audience is bypassed so it is easier to perform and mass produce. This is the method used by the vast majority of films, books and plays, and this is also the method that is most often used in the current range of interactive computer games and fiction. While some adaption is made to give the audience some interaction (as shown in Section 2.3), the story itself remains within the control of the designer.

While this method is capable of dramatic and deep storytelling, as proven by its success in the traditional storytelling media, it has a number of shortcomings:

- With no method to significantly influence the plot, there is a tendency for the audience to feel powerless. If the interactivity does not have a significant effect in the path of the story, then the choices made by the audience cannot be truly important.

- Since this model is the one used by the traditional media of books and film, then for the computer to also just offer a linear plot means to simply compete against the current storytelling formats. The advantage of using the techniques perfected in other media is a definite disadvantage when the product is vying for the market. It also begs the question of whether the medium of computerised storytelling is worthwhile if stories of this kind can be better told in other media.

- Finally, since this form of storytelling is not truly interactive, it means that the art form is not meeting its full potential. There is the opportunity for a different type of storytelling with computers, and it diminishes the format of interactive storytelling if the possibility of interactive storytelling is dismissed. Storytelling by this method is analogous to the player on the stage acting from a pre-written script (such as by autocue).

This approach makes the storytelling technique an adaption of the traditional fixed scripted style. The advantage is that it is makes the storytelling easier for the actors, as the plot is guaranteed to perform down the path they have prepared. However, it has the potential to alienate and frustrate the player, as there is no way the player can influence what is going to happen to their character, which is effectively themselves (as their character is identified with themselves[8]). This might not be a problem if the player agrees with the path the story is taking or the decisions their character has to make, but which paths are considered agreeable will vary from person to person.

While this method can be (and is) used for computer storytelling, the resulting story is similar to that produced in the existing media. If the story could be equally well-told as a book or a film, then it is missing the potential of the computer medium.

Giving the audience all the power to shape the story makes them the masters of the narrative. The nature of the storytelling experience is up to them, as they are the directors of the story. In this case, the player is the story-teller and the actors follow their instructions. This is similar to a child playing with a doll-set, except the dolls are real characters able to improvise their respective roles. This would give the player a great sense of empowerment, as they would have total control over the story. However, the resulting story would be only as good as the player could make it, and

[8]The nature of the interactor's representation as the protagonist in interactive storytelling and the problems that this creates for interactive storytelling systems is explained in Section 5.2.1 later in the dissertation.

there is the danger of the player feeling unchallenged by this method (as if you have the power to do everything, then there is no real accomplishment in achieving anything). There is also the practical problem of scale; there must be some limit as to what actions or situations are possible, as there is only a fixed level of competence that an actor or world has in an interactive computer environment. However, this is still a viable way of providing a story-telling environment, as the best-selling game *The Sims*[32] has shown.[9]

While the other methods are used successfully in current interactive storytelling, the final method, sharing the control between the audience and the system, has yet to be properly implemented. With this system, the desires of the audience towards the path of the story must be ascertained by the storytelling system, but the system still maintains the role of storyteller. Although it is still the responsibility of the system to tell the story, the system still needs to pay attention to the wishes of the audience and mould the path of the story to fit their expectations. This is the full potential of interactive storytelling. To date there has not been a software system that has achieved the ideal of interactive storytelling, so a working example that has been developed for the computer such as those for the first two approaches cannot be given. However, a good example often given by game designer and interactivity expert Chris Crawford is that of a parent telling a child a story.

> The parent has in mind a story to tell including what characters it will involve, what surprises it will contain, roughly how the story will unfold, and approximately how it will end. But as the child asks questions about the story, the parent will change the

[9] At time of writing, *The Sims* was the best selling PC game.

story accordingly. The parent may use a book as a guide, but will stray from that guide as necessary. For example, the story might begin:

"As the princess wandered through the dark forest, she was frightened by many different things she saw, including a large newt, a dark cave, and an old shack."

As the parent tells the story, the child may ask questions.

"What colour was the newt?"

"The newt was a strange shade of yellow, a colour the princess had only seen in the royal spiced mustard."

"What about the cave?"

"From within the cave came a terrible smell, reminiscent of the smell of sulfur burning."

"Maybe there's an old sorcerer in there, making potions. Does she go into the cave?"

"She did enter the cave, taking each step carefully in order to avoid stumbling in the dark. And as she went deeper into the cave, she started to see a light, and a voice shouted, 'Who is it that enters my cave?' And as she got closer, the princess saw an old wizard with tattered robes..."

There may not have actually been a sorcerer in the story as the parent had initially intended to tell it, but as the child asks questions, instead of answering "you can't go that way" or "there's

> nothing special about it" as a poorly designed computer game might, the parent adapts the story to the child, adding detail and introducing new characters and situations as necessary. The overall story arc and its main protagonists may not change much, but the child has had a real role in determining what exactly happens in the story.
>
> Richard Rouse III, game designer and author, on Chris Crawford's example of interactive storytelling[42, p.217]

Of course, the parent in this example has the advantage of a lifetime of experience with stories (as well as general experience from living through everyday life that should also be used by a human storyteller). A human storyteller also has a much better ability to listen to their audience, and to be flexible enough to cleverly adapt their story. Without the benefit of a human-level intelligence, special methods need to be designed to automate stories so that they can be encoded into software. Presently the best that can be done is to anticipate the actions of the audience in advance; the human authors of a piece of interactive fiction can guess what the audience will do, where they will go and which choices they will make. However, as the size of the story grows, so does the amount of work that the authors must do. Using this approach, the amount of preparation needed actually grows exponentially compared to the size of the story[10]. Thus this approach is not viable as a general solution for interactive storytelling of any substantial depth.

Truly interactive storytelling that offers choice to the audience cannot be

[10] This is explained in more detail later in the dissertation, in Section 2.3.

provided using the same methods of writing that are used for linear media. To solve the problem of the exponential growth of the work required by the authors of interactive storytelling using traditional linear methods, an algorithmic approach to storytelling is required. The system will need to work with a higher level of abstraction of stories; they will need general representation of story structure in order to cope with the choices of the audience. Thus this representation needs to be able to react to the decision of the interactor of the system while able to maintain a sense of structure. It is the aim of this thesis to argue the feasibility of the design of such a storytelling system, show such a representation, and to explore how this can be used to develop an interactive storytelling system.

1.1 Major contributions

The following major contributions to interactive storytelling have been developed and are presented in the thesis:

- An abstract representation for interactivity within story structure, the door and key model, which is capable of algorithmic plot generation suitable for storytelling systems;
- A prototype of the door and key model that has been shown to generate story structures analogous to those proposed by storytellers and analysts such as Livo, Rietz and Propp[25, 41];
- An analysis of story mappings and structure developed for storytellers, writers and folklorists with respect to their applicability to the field of

interactive storytelling.

1.2 Overview of thesis

In this chapter the problems of interactive storytelling have been outlined and the goal of the thesis stated: to explore the aim, needs and problems with interactive storytelling and develop a method of representation that can be used for interactive storytelling systems. This section will briefly describe the overall structure of the remainder of the thesis.

Chapter 2 describes the nature of interactive storytelling; covering the definition of interactive storytelling, the concepts of interactivity when applied to this domain, and the existing methodology used with the present interactive fiction and computer games available commercially.

Chapter 3 covers models of story structure developed by theorists in the field of literature and story writing, and how the constructs prevalent in different genres of storytelling can be used in the development of interactive storytelling systems.

Chapter 4 details the related work done on computer-based interactive storytelling; modelling story characters, and story writing systems.

Chapter 5 outlines the theory behind an interactive storytelling system, giving the requirements of such a system from the respect of system design, and describes the abstract 'door and key' model for interactive storytelling.

Chapter 6 gives some sample output of the 'door and key' model, and shows how this relates to the current interactive stories (in the form of computer adventure games) and to the literary theories of story structure, showing how this model can be used as the basis of a full interactive storytelling system.

Chapter 7 discusses conclusions from this thesis and outlines the future directions that work on interactive storytelling systems needs to address.

Chapter 2

Nature of Interactive Storytelling

> Imagine all of the power of a story in a novel, with its ability to grab hold and captivate the reader, to make her care about the characters in the story, to change her perception of the world, and, in some special instances, to change the way she lives her life. Now imagine how much more powerful that would be if, instead of reading about the actions of other characters, the reader was the main character in the story and was able to make choices that would affect the shape, direction, and outcomes of the story.
>
> Richard Rouse III, *Game Design: theory and practice*[42, p.215]

2.1 Definition of interactive storytelling

There is a lot of potential for confusion over terminology in such a relatively new area as interactive storytelling or other forms of computer arts. Quite often critics and analysts of electronic media, deprived of an appropriate lexicon of terminology, have to invent their own terms, sometimes leading to some confusion about the meaning of specific terms. Over time standard definitions of the terminology will arise from consistent usage, but for this thesis important terms will have to be explicitly defined.

The term *interactive storytelling*, used in this thesis, is not as widely used as other similar terms in the field. This is intentional, as the concept of truly interactive storytelling has not yet been fully realised. Since the terminology that is used, such as *'interactive fiction'* or *'interactive drama'*, already have developed a standard definition from their use. The use of the term *interactive storytelling* instead of *interactive fiction* is intentional, as the established definition of interactive fiction is a quite separate concept.

Interactive fiction is the term used to describe the class of computer games that used to be known as *adventure games*, a genre named after the first game of its kind, *Adventure*[12], but which now has had its meaning diluted somewhat by the marketing of action and skill based games with a backstory as 'adventure'. The interactive fiction genre of games, ranging from the purely text-based originator *Adventure* and *Zork*[2] to the graphical successors (such as Sierra's *King's Quest* series and Lucasarts' *Monkey Island* series[37, 7]), contain distinctive traits that differentiate them from other genres. Interactive fiction has a story at its core, whether simple (as in the 'retrieve the treasure' goals of early interactive fiction) or complex (the epic quest and detective stories are popular themes for interactive fiction). The key interactions (or 'gameplay') that are the hallmark of interactive fiction is the use of environmental puzzles for the interactor to solve. These puzzles are initiated with the presence of an obstacle that hinders the progression of the story, which must be overcome by clever means (frequently by using an inventory of items the interactor has found along the way). While the genre of interactive fiction is a worthy component to the halls of electronic media, these games do not fit the criteria that this thesis is based on.

Other terms, such as *interactive drama* are used by researchers in the field of virtual storytelling or computerised characters. These terms are used to describe a range of different interactive situations, the scope depending on the researcher's own definition. An example is with Nicolas Szilas' definition of interactive drama as "drama on computer where the user *is* a character. *Being* a character means being able to perform any action on the fiction world that the other characters can perform."[59] These projects are usually highly character driven, with a focus on the development of realistic characters or expressive character interaction. While these advances in realistic characters are a useful component of a storytelling system, this is a different aim from a truly interactive story system.

There exists considerable overlap between interactive fiction and interactive drama; however in order to distinguish the aim of this thesis from the existing interactive fiction, and from related but different goals of other researchers, it is important to use terminology freed from any pre-established connotations derived from its prior use. As such this thesis makes a distinction between these forms of interactive drama by using and defining the term 'interactive storytelling'. The obvious (and somewhat trite) definition of this term is obvious and in fact true; interactive storytelling means telling a story interactively, which is a suitable reason alone for the choice of these two particular words. However the meaning of the two words of 'interactivity' and 'storytelling' and how they relate to each other need to be ascertained. In order to gain an understanding of storytelling that is capable of being interacted upon, it is necessary to explain the fundamentals of both storytelling and interactivity. By studying the fundamentals of these two components, the nature of interactive storytelling can be understood. The important aspects of storytelling are included later in the thesis in Chapter 3. However the core aspect of this system that differentiates it from the story generation models described in Chapter 4, and from the existing systems today, is the concept of interactivity.

2.2 Interactivity

> There's a conflict between interactivity and storytelling: Most people imagine there's a spectrum between conventional written stories on one side and total interactivity on the other. But I believe that what you really have are two safe havens separated by a pit of hell that can absorb endless amounts of time, skill, and resources.
>
> Walter Freitag, *Juul (1996)*

'Interactive' is an adjective that is widely used these days to describe almost every electronic device that responds to input from a human. It has become a buzz-word that seems to be used almost automatically for any computer application. However, despite its frequent use over the last few decades to describe all manner of computer applications (including storytelling, as can be seen by the frequency of its use in the titles of the papers in the bibliography of this thesis), interactivity has been open to various different methods of interpretation, leading to some confusion as to what it exactly means.[1] Therefore in order to clear any ambiguity about such a debated term an appropriate definition of 'interactivity' needs to be stated.

Due to the modern usage of the word in the domain of software, interactivity could be defined to describe any computer application with an input. This is what some authors have done with their books on interactivity, and it is starting to be recognised as a technical term in dictionaries. For example, *The Macquarie Dictionary* gives two definitions of *interactive*:

1. *adj.* of or pertaining to things or persons which act on each other.
2. *Computers.* (of systems, etc.) immediately responsive to commands, data, etc., as opposed to systems arranged for batch processing.

This second definition is the one that is usually implied when the term 'interactive' is used in reference to any computer application. However this particular definition of interactivity is not very useful when used as a requirement of an interactive storytelling system, due to two problems implicit

[1] For example, Laurel writes how at the 1988 INtertainment conference, the first annual conference of a variety of interactive entertainment businesses (including personal computers, video games, broadcast and cable television, optical media, museums, and amusement parks), the debate on the definition of 'interactive' spanned the length of the conference, with no speakers presenting a definition that gained general acceptance.[22, p.20]

within the definition.

The first problem arises from the definition being a binary entity; either something is interactive or it is not. As long as the system responds immediately to commands, regardless of the range of commands possible or how it responds, it can be called interactive. If used this way for interactive systems, it becomes a hurdle requirement that is either passed or failed. It also means that two systems cannot be compared on their level of interactivity; it becomes impossible to say that one system is *more* interactive than another.

The second problem stems from the binary nature of the first, and is evident in its common use with interactive fiction today. Defining 'interactivity' as any computer system that responds to commands means that if any aspect of the system qualifies as interactive then the whole can be labelled as such. Thus the interactive fictions, stories and adventure games tend to be constructed of interactive puzzles or action sequences, but at their core they have a static linear story.[2] This use of interactivity to describe most commercial software has made the use of the word redundant, as all software that people typically encounter in their everyday lives when using computers could be described as interactive using this definition. Since this level of in-

[2]This was most evident during the era of computer games around 1993-4 that arose when CD-ROMs became available to the general consumer. Many games were released that were marketed as 'interactive movies', which unfortunately gave an experience that consisted of a series of overly long film clips, generally starring low grade actors, separated by simplistic puzzles that were the extent of the products interactivity (examples include the Sega Mega CD title *Night Trap*, and Sierra's PC game *Phantasmagoria*). The consumer reaction to these products ensured that the genre did not have a long commercial lifespan. This is an example of how the use of 'interactive' as a marketing term has diluted its meaning.[49]

teractivity is too simple a requirement for an interactive storytelling system, and has thus already been achieved by existing systems (all computer games that contain a plot could claim to qualify), it is not this definition that is intended by ‘interactivity’ by this thesis, or by the other researchers in the field.

A better definition of interactivity, and the one implied by its use in this thesis, arises from the first (and original) definition given in the dictionary as listed above; that of things or persons which act upon each other. This definition, derived from ‘interact’, puts a different perspective as to what constitutes an interactive computer system, if you consider the system to consist of an human interactor and a computer acting upon each other. From this definition, in an interactive system there should be effectively an equal relationship between the parties involved. Hence for an interactive system both the computer must react to the human, and the human to the computer. The analogy used by Crawford is of a conversation between two equal parties. One party speaks, and the other listens. The listener must then respond with something pertinent to what the speaker has just said, and thus the roles are reversed. For a proper conversation, both parties must be contributing, and both must be responding to each other. This is a good analogy for how a proper interactive system should behave.[10]

The ‘conversation’ model of interactivity is also useful in that it turns the term into an analogue concept. It is possible to say one system is more interactive than another by the extent to which both parties can respond to each other. For example, a film shown in a theatre cannot really be considered interactive, as the only input a viewer can have is to leave. A film shown on a DVD can be considered more interactive still, as it gives a viewer the power to rewind or fast forward, skip scenes, zoom, and choose subtitles and commentary. However it is readily apparent when comparing all of these to a proper conversation between two people that the viewer still has little say about the path of the film, and so all these examples would be considered at a low level of interactivity. Even with this analogy driven definition the concept of interactivity is still an extremely fuzzy concept (for example it would be difficult to devise a numerical interactivity score). It is still a useful metric for assessing the relative interactivity of different approaches.

The conversation analogy also works well when considering interactive storytelling. For a truly interactive storytelling system, both the storyteller and the audience must react to each other's input as like a conversation. The audience must be responsive to the story that is being told, and the storyteller must allow the audience to have their proper input into the path of the story. There must be a partnership in the path of the story, with the audience having an active role in the path of the plot.

Janet Murray defines agency as 'the satisfying power to take meaningful action and see the result of our decisions and choices'[35, p.126]. She uses the term *agency* to describe this nature of interactivity.

this is to help describe the confusion between interactivity and mere activity. For example, a board game of chance can involve players keeping busy, spinning dials, moving pieces, exchanging money and tokens, but have no control of the action, i.e. no agency. However, a slow paced game of chess involves choices that have dramatic effect on the path of the game. The number of interactions per minute in a system is a poor indicator of agency[35, p.128]. This confusion over agency and activity is common in electronic entertainment and computer games, where applications are described as 'an interactive story' because they contain a story and interactive elements. It must be stressed that unless the interactivity applies to the plot of the story itself, it does not satisfy the objective of an interactive storytelling system.

Note that this definition of interactivity with respect to storytelling is not unique. This definition is similar to the definitions used by other researchers and commentators of interactive storytelling, such as Laurel[22, p.20–21], Murray[35, p.70, 126], Stern[55, p.16], Szilas[58], and Tichenor[61].

.

2.3 Existing methodology

One of the main arguments of this thesis is that true interactive storytelling, in the sense of giving true agency to the audience regarding the control of the path of the story, is not yet evident in the current example of electronic entertainment and interactive fiction. The reason behind this lack of true interactivity in the electronic entertainment titles present today lies in the fundamental methodology used to encode the stories for use in the computer. In this section the techniques used in computer games are summarised to show why this is so.

Currently the most common form of electronic interactive entertainment is provided via video or computer games. The titles available cater to a wide spectrum of tastes; arcade twitch games, puzzles, strategy, simulation are a few examples. Many of these games do not feature any storytelling, and some would arguably not benefit from its inclusion (for example, abstract games such as the popular tetronimo stacking game *Tetris*[38]). However most of the current titles feature some kind of plot, ranging from sweeping dramatic epics akin to those found in good literature or cinema, to the flimsiest of storylines present purely as a device to stop the game descending into meaningless chaos. Some types of games (such as many in the action, strategy or puzzle genres) are not suitable examples of the present state of interactive stories as that is not their goal. However there are many titles that rely on strong characterisation and plot, such as those that classify themselves in the adventure and role-playing genre. For these titles, the story is usually a core component in the enjoyment of the game, and is one of the big selling points for these types of games. It is these titles that are regarded as the best current examples of commercial interactive storytelling.

While the existing range of computer games provide some excellent examples of storytelling (such as the award winning and dramatically compelling *Grim Fandango*[8]), they are not very good at telling *interactive* stories. With regards to the structure of the plot, games have not altered much in the last decade. Nearly all of the stories in games can be represented by one of the following graphs or tree structures, where each node represents a state in the story. These trees are similar to that used by Crawford in an article in 1993[9]; they are still applicable now.

A typical linear story, such as one in a book or a film, is shown in Figure 2.1 would be a simple series of states. One event follows another, no variations are allowed, so therefore this is obviously not interactive.

The simplest modification to turn this type of story into interactive expe-

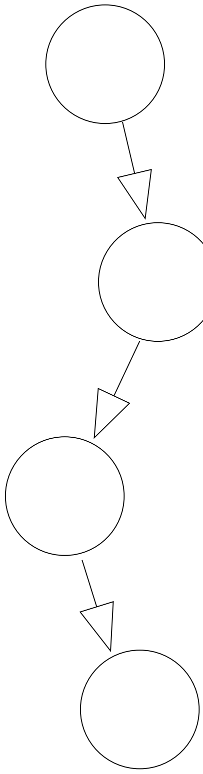


Figure 2.1: Linear story tree diagram

rience is to splice some challenges or puzzles into the tree (shown in Figure 2.2). Successful completion of the challenge is required for the story to continue to the next state. Despite its simplicity, this method is widely used in a large variety of games; the series of challenges (such as ‘missions’ or ‘levels’) provide the gameplay and the story acts as the hooks that give some sense of cohesion to the whole ensemble. However, while the challenges themselves may be interactive, the story itself is not.

To actually include interactivity into a story the person playing the game needs to be given the sense of agency; they need to be given decisions that

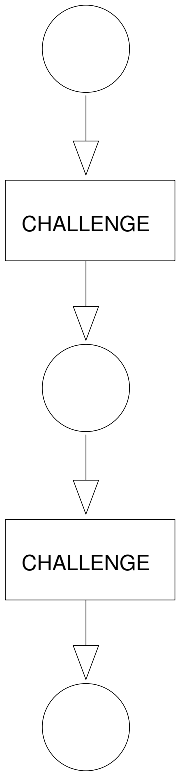


Figure 2.2: Linear story tree diagram with additional interactive challenges

affect the path of the plot. A pure form of this game is represented in Figure 2.3; at every stage the player is given a meaningful choice which uniquely affects the path of the story. In this example each choice is the simplest that is possible; a binary yes or no decision. In theory a game developer could write each state of this tree as per the linear methods, and thus end up with an interactive story. However in practice no games of any substantial size could do this, as the amount of work required to implement this approach grows exponentially with the size of the story; as such this approach is unworkable for any reasonable sized story. The tree shown in Figure 2.3 is only three levels deep, and as such any one path from the start of the story to the finish involves only three decisions from the player. To allow four decisions per path, one more level to the tree, the game developers would need to write double the number of states. For every extra decision added, the number of states needs to be doubled. To have thirty unique decisions, which would be a reasonable number for a short interactive story, requires the creation of over a billion states. This exponential growth in the number of pre-written states is why it is not possible to create in advance a unique state for every combination of decisions.

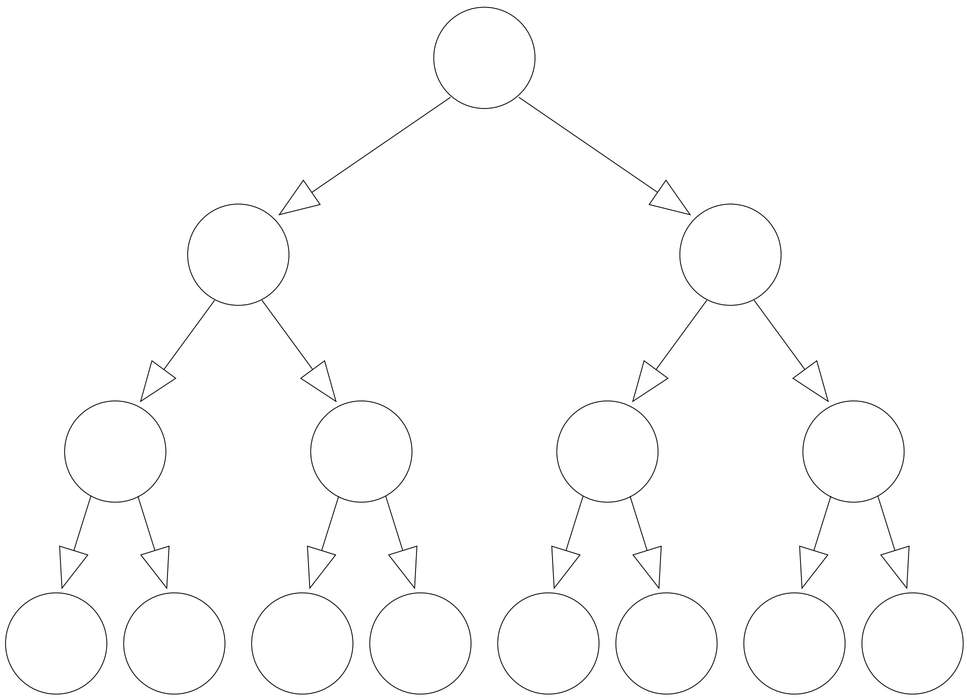


Figure 2.3: Interactive story tree with two decisions at each branch, three levels deep

In order to build an achievable interactive story techniques have to be implemented to remove the need to pre-design unique states along every path of the story tree. One technique is 'foldback', where divergent paths of the tree eventually return to each other. This will slow the exponential growth, but it will not solve the problem. If there is a lot of foldback (so the number of states at each level does not increase) it robs the meaning of the choices made by the player; whichever decision they make, the same state will be eventually reached. If there is not a lot of foldback (so the number of states slowly increases), there is still exponential growth, so the problem is not solved.

Another technique (and one frequently used in adventure and fantasy role playing games) is to thread multiple smaller stories together, where each thread can be advanced separately. In adventure games, these threads are often sub-problems which all need to be successfully traversed in order to 'solve' the game. In role playing games, these are often optional storylines with little interrelation (the threading allows them to be easily edited by the developers). This may change the flow of the story, but it does not affect the problem of exponential growth.

The problem with the stories in existing commercial games is that they have to be hand-written. The large number of decisions that need to be made in an interactive story lead to an exponential amount of work needed to build it. There is therefore a tendency to want to limit the interactivity in order to alleviate the workload. There is also commercial and artistic pressures. Commercially, it is hard to argue that so many 'extra' scenes need to be written that the player will never see. Artistically, there is a sense of familiarity and style with the traditional linear storytelling that many game designers like to implement.

This review is not a criticism of the existing methodology for stories in computer games, as the moving stories in some of the landmark games are of great dramatic value. However the interactive form of storytelling is a different genre made possible to mass market due to the presence of computers. This argument is not to disrepute the dramatic integrity of the existing form, but to show how the techniques used to write these works cannot be used to create the interactive form. Traditionally hand-writing the story cannot be effectively used to build truly interactive stories due to the exponential growth problem. So other techniques need to be developed to tell interactive stories. These include automated story building and advanced character modelling, techniques of which are described in subsequent chapters.

2.4 Conclusion

> In essence much effort has been put into proving the following: If narratives are linear stories written without the presence of the interactor then narrative and interactivity are opposite categories. This is truth by definition.
>
> Jonas Heide Smith, *The "how's" and "why's" of interactive fiction*[50, p.21]

In this chapter the nature of interactive storytelling and the fundamentals of interactivity with respect to the domain of stories was reviewed, with an explanation why writing for computerised stories will not work in interactive environments using the existing techniques of encoding the story as a series of pre-written states or scenes. It is only possible to gain a sense of agency using these techniques if given boundless time and resources. While appropriate for fixed story-lines this methodology does not scale well to large interactive stories, as the number of required states that need to be written increases exponentially proportional to the length of the story. It is not possible to sidestep the problem by getting a human writer to design every scene. Any solution to this problem must involve some level of automated story generation.

In the next chapter, the fundamentals of storytelling, story structure and rules for generating plot are given, using techniques designed by storytellers, scriptwriters and folklorists. These aspects of storytelling are analysed with respect to their applicability to an interactive domain.

Chapter 3

Storytelling

3.1 Nature of stories

> A plot is ... a narrative of events, the emphasis falling on causality. "The king died and then the queen died" is a story. "The king died and then the queen died of grief" is a plot.
>
> E. M. Forster, *Aspects of the Novel*

It is important to understand what constitutes a story. As noted by Murray, E. M. Forster got it wrong; "The king died and then the queen died" is not really a story; it is a chronicle, a sequence of events. Furthermore, it takes more than causation to turn a chronicle into a rich plot[36]. It is not enough for a story system to merely generate events, even if they are given some sense of purpose.

The nature of stories is a difficult one to explain. People gain an instinctive understanding of what constitutes a story from a very early age[25, 30, 54, p.28]. Due to this subconscious understanding of the nature of stories that people gain through experience with these constructs it becomes difficult to give a concise and accurate definition that is inclusive of all articles that are considered stories and exclusive of those that are not. Since giving a proper definition of a story is difficult, many researchers avoid giving an exact definition of a 'story', relying on the instinctive understanding that everyone has of stories to negate the need for any explanation. While this trusts the definition to an intuitive form of common sense, it is suitable for a general requirement of interactive storytelling systems. However, it is important when gaining an understanding of the goal of interactive storytelling to realise that the presence of interactivity must not negate the property of the experience being a story. This is the reason why interactive storytelling is different from interactive drama, puzzles and other experiences; as these need not be tied to the traditions of the forms and structure that is implicit to storytelling.

This thesis focuses not on stories but on interactive *storytelling*. There is a reason, further than merely distancing such systems from the preconceived notions that arrive from the current situation, for choosing to describe such systems as 'interactive *storytelling*' rather than 'interactive *fiction*' or even 'interactive *stories*'. The terms stories and fiction imply those kinds of media that we are experienced with, such as books and film. Despite their history and heritage as forms for fiction, these are static media, and the stories told in such forms are not able to be adapted. Storytelling, in contrast, implies the existence of a storyteller who is able to adapt the story to fit the target audience, much like the storytelling bards of ancient times. It is an oral tradition, and as such involves a sense of malleability to the story, rather then the fixed and immutable nature of stories captured in print or on film. In oral storytelling it is expected that some level of improvisation or adaption of a story will be made by a particular storyteller or singer[26]. It is this adaptability of the story, the ability to change the nature of the story for the audience that is the key part of an interactive storytelling systems. For the system, the integration of interactivity is a core requirement for the model of stories used. Therefore it is useful to look at the existing forms of storytelling that could truly be considered interactive, such as the works of the oral storyteller.

3.2 Oral storytelling

Storyteller:	I'm going to tell a story.
Audience:	Right!
Storyteller:	It's a lie.
Audience:	Right!
Storyteller:	But not everything in it is false.
Audience:	Right!

Sudanese ritual opening[25]

Since this thesis is discussing the plausibility of interactive storytelling, it is fitting to look at the originator of all forms of story based fiction, that of oral storytelling. The tradition of oral storytelling extends into pre-history, being the method for storing and transferral of oral literature and culture before the invention of writing. Despite being an ancient art, there is much to relate oral storytelling with the potential of interactive storytelling (as stated earlier, this is why this thesis uses the term of interactive *storytelling* rather than interactive stories or fiction.). Since we are considering electronic storytelling it is wise to consider the human analogues.

First a clarification needs to made; in this thesis *oral storytelling* does not refer to the reading aloud of books, such as is the practice often used to tell stories to groups of children in libraries. While still an oral form of presentation, this is still essentially the same delivery of the story as if the book itself was read silently. Instead, this term refers to the practice of storytelling used by the classic storytelling bards and poets (such as the epic bard Homer), and while diminished in its universal popularity in modern western society is still in use today[1]. These oral storytellers do not recite from written material (in fact the art seems most practised by those who are illiterate[26, p.20]). Instead they must rely on their memory, using techniques that are a good source of understanding for the design of an electronic storytelling system.

With oral storytellers, it is important to understand how they are able to remember their stories. Albert Lord sought to find this out in order to show that the Homeric poems that form the classic *The Iliad* and *The Odyssey* came from an oral source[26, pp.8–12]. Some literary experts thought that the Homeric poems were penned by a single author. However those working with the poems found many problems with this opinion. One problem was

[1] Although in western society oral storytelling is now associated with reading to children[25, p.8] in some societies the traditional storytelling bard is still a prominent part of their lifestyle (such as the singing poets of former Yugoslavia studied by Albert Lord[26]), and indigenous peoples that preserve their culture through traditional oral storytelling methods (such as Paula Underwood with Iroquois history[52]).

whether writing existed in the ancient Greece of what was thought to be Homer's period of (pre-)history (about the ninth century B.C.). Another was the inconsistencies in the Homeric poems; this was combined with the presence of different dialect forms. A further problem was the unusually long length of the poems; as if those of Homer's period in history did not have the ability to write then it was puzzling how these poems be preserved for prosperity, or indeed how they could be written down at all[26, p.8]. These problems are solved if the poems were being passed on as a traditional tale among oral storytellers. With this opinion on epic poetry, they must have been created by no single author, but by the series of poets throughout the ages who told and refined the tale through the generations. However this brings up the question of how these oral storytellers could remember something as long as *The Iliad*. There are still those who practice the singing of poetry, as thought to be the style used by the oral storytellers in the age of Homer. Lord's research into the singers of (what is now former) Yugoslavia came up with some analysis into how these storytellers manage to achieve this feat. In the following paragraphs, relevant aspects of Lord's book, *The Singer of Tales*, are summarised[26].

The bards and poets do not remember their stories by rote memorisation. An experienced oral storyteller would know many dozens of stories, and to remember each story word for word would require an outstanding memory (furthermore, to remember the entirety of an epic poem such as *The Iliad* perfectly would require a exceptional person, and there are too many storytellers for this to be true). Also, Lord found that a specific storyteller, when retelling the same story at a later date, would have significant differences in their content[26]. It is also clear that the oral storyteller is not improvising each story from scratch every time, as otherwise epic stories such as the Homeric poems would not be passed on to the modern day.[2] Oral storytellers learn from copying each other, and it would be impossible for a storyteller to know any 'favourite' stories of their intended audience if no story memory was used. However two different storytellers are able to tell what is recognisably the same story. Obviously the storytellers must have a way of remembering the stories. In fact what Lord found was that the storytellers have made a compromise between memory and improvisation; the storytellers could remember the core aspects of the story, but improvise the specific words sung. The reason the oral storytellers could achieve this lies in the fundamental nature of the storytelling.

Core to oral storytelling is that it is heavily based on underlying rules. Honed through generations of tradition, each culture devised its own set of rules for their set of stories. These rules govern all aspects of the story, from ritualistic beginnings and endings (for example the 'Once upon a time' and 'they lived happily ever after' of Western European fairy tales), routines for audience participation (for example the 'Look behind you!' used in pantomime and puppet shows, or repetition stories used for children) to the types and use of the story structures and supporting characters contained within. The rules form the framework around which to construct the story, and apply to both the audience and the teller as to what their roles should be. Thus, as written by Livo and Rietz, "Storytelling is an organised, con-

[2]This also would violate another important use the oral stories have - that of maintaining the cultural memory of pre-literate societies[25, p.8]

trolled, rule-governed storying of information, using the story shapes of the oral literature, the people, and the medium of the live storyteller."[25, p.8] The acknowledgement of the rule and pattern based nature of oral storytelling makes this particular art form attractive to the design of computer storytelling systems.

The reliance on the rules that is the cornerstone of oral storytelling may seem to be disadvantageous to its creative potential, as any form of structure or rules provides some level of limitation. In written literature it is possible for the author to wilfully ignore the conventions of genre, to turn tradition on its head, and become the stronger for it (as shown in Laurence Sterne's *Tristram Shandy* written in the eighteenth century, in books time can jump around and whole pages can be left blank to transcend the perceived notions of the use of the medium[56]). The imposition of rules and patterns in oral storytelling are vital to the storyteller as they provide the framework of ritualised components from which to easily build a story. As such the rules and patterns themselves become what defines these genres of stories, and become a crucial component necessary not only as an aid for the storyteller to remember the story, but as a critical part of what defines the medium itself. The patterns, rules and rituals combine to form the whole experience that is storytelling; without them, it would not be a 'story'[25, pp.28–30]. The strength of these stories and the allure of storytelling as a medium is not diluted by this fact.

As stated earlier, the rules of oral storytelling are specific to certain styles and cultures. Some of these cultural differences can be seen in the ritualised beginnings (or endings) to stories, such as Sudanese beginning shown at the beginning of this section. Others relate to the specific style of the performance, such as the use of ten syllables followed by a syntactic pause in the Yugoslav tradition of singing tales[26, p.32]. While these story rules and traditions are an important part of the overall performance of a storyteller, it is not critical for this thesis to delve deeply into analysis of the fine word crafting rules that are used by storytellers to craft the individual sentences of their stories, as these elements are the fine detail in the craft of the story. The backbone of any computer storytelling system will derive from the rules and patterns for the major elements of the story: what can loosely be called the ‘plot’. There are two main attributes of the patterns in the stories told by oral storytellers that are useful for analysis: the *story structure*, and the *archetypes of characters.*

3.3 Story structure

In this section, existing models of story structure developed by storytellers and folklorists are presented. While this models were designed for authors, storytellers or folktale classification, here they are examined for their suitability for adaption to the area of interactive storytelling.

3.3.1 Story mapping

Hero stories are based around *problems.* At the centre of the plot the protagonist of the story has (at least one) problem to solve. The body of the story consists of a series of *events* where the protagonist attempts to solve the problem (during which new problems may be created), The story ends when either the problems are solved, or the solution of the problems are no longer relevant (due to, for example, the death of the protagonist).

Norma Livo and Sandra Rietz's guide to storytelling (*Storytelling: Process and Practice*[25]) gives the following outline of how a storyteller might map the general structure of a story into their memory, in which the major sections of a story are broken up into separate 'slots' for the teller to remember.

1. **Introduction to setting and characters: development of the problem**

   The development of the problem might also consist of a small series of events. If so, the introduction can be broken down further:

   (a) Problem establishing event 1

   (b) Problem establishing event 2

   (c) etc.

2. **Problem**

   Clarification/statement/description of the problem.

3. **Event sequence**

   In the event sequence, the protagonist or protagonists undertake a series of actions designed to solve the problem. Sometimes, these activities lead directly to the problem solution in a straight, linear fashion. A direct, linear problem solution might map as follows:

   (a) Event 1 - leads logically to the next event

(b) Event 2 - leads logically to the next event

(c) Event 3 - leads logically to the next event

(d) etc.

(e) Event second to last - leads logically to the last event (the problem solution)

Alternative event sequence

Another possibility for an event sequence is the ‘embedded’ solution, where the protagonist’s attempt (or attempts) to solve the original problem only ends up creating more problems. Often these additional problems are cumulative - problem 1 creates problem 2, which creates problem 3, which creates problem 4. Then the problems must be solved (usually in reverse order).

(a) Event 1 - attempt to solve problem 1 instead causes problem 2

(b) Event 2 - attempt to solve problem 2 instead causes problem 3

(c) Event 3 - attempt to solve problem 3 instead causes problem 4

(d) Event 4 - solution of problem 4 allows return to problem 3

(e) Event 5 - solution of problem 3 allows return to problem 2

(f) Event 6 - solution of problem 2 allows return to problem 1 - which leads logically to the last event (the problem solution)

This event sequence involves a cause and effect relationship between events, where each new problem is embedded in its predecessor. Since each problem is the effect of the preceding action it must be solved before the preceding problem can be addressed.

4. **Resolution (problem solution)**

   Since the last event is sometimes organised differently than its predecessors (it may contain a slightly different content or a different arrangement of content) it might be considered to be a special ‘slot’ separate from the sequence of events.

5. **Conclusion**

   The conclusion provides the teller with a place for the wrap-up of the story

6. **Moral**

   Some stories add a formal moral at the end of the conclusion.

(Adapted from Livo & Rietz, *Storytelling: Process and Practice*[25, pp.33–35])

Using this story map, a storyteller would slot the story-specific content into the appropriate sections. The types of events used depend on the nature of the story, and usually involve a level of repetition. Here is a summarised children’s story example used by Livo and Rietz to explain the use of the story map, printed here close to the original in a summarised (but still quite amusing) form.

THE WIDE MOUTHED FROG[3]

1. **Introduction to setting and characters: development of the problem**

   Wide-mouthed frog lives with wife along riverbank. (Details about home, surroundings, etc., are given here.) Wife about to have babies - but what to feed them? Husband agrees to find out, and sets out from home on a quest for an answer to the question.

2. **Problem**

   How to find out what to feed wide-mouthed frog babies.

3. **Event sequence**

   (a) Frog meets mouse.

   Frog says, "Hello! I'm a wide-mouthed frog. What do you feed your babies?"

   Mouse answers, "Grain, cheese. Steal it."

   Frog responds, "That's probably not a good thing to feed to wide-mouth frog babies. Thank you."

   Frog hops on.

   (b) Frog meets snake.

   Repeat entire event for snake.

   Substitute snake answer: "Raw eggs. Steal 'em."

[3]Original author of this particular version of this story unknown to Livo & Rietz[25, p.35]

(c) Frog meets owl.

Repeat entire event for owl.

Substitute owl answer: "Live mice. Swallow 'em whole. Spit out the bones."

(d) Frog meets lion.

Repeat entire event for lion.

Substitute lion answer: "Red raw meat. Chew it all down."

(e) Frog meets dragon.

Insert description of dragon and dragon activity (roasting marshmallows on a stick using own breath, pulling them off gooey, eating them, teeth sticking together, etc.)

Frog calls up to dragon.

Dragon answers down to frog, "I feed my babies wide-mouthed frogs."

(Embedded problem - if the dragon discovers that the frog has a wide mouth, the frog might be fed to the dragon's babies.)

4. **Resolution (problem solution)**

Alternative 1 - Dragon catches (and eats) frog.

Alternative 2 - Frog narrows mouth to avoid detection.

Livo and Rietz note that in this version of the story the frog does not solve his original problem - what to feed his own babies. Only the embedded problem is solved (in alternative 2) or made redundant (if the frog is eaten in alternative 1). With alternative 1, young audiences often fail to get the joke, and do not like to see the frog get eaten (especially if the storyteller has been effective making the frog an endearing character, with children chanting along with the frog throughout the story). With alternative 2, children often fail to understand the significance of the narrowing of the frog's mouth at the end of the story. They are waiting for the conventional ending in which the frog finds out what to feed his children and goes home to report to his wife. Since this does not happen, children often suggest alternative endings for the story in which the frog gets his answer and goes home. This is an interesting example of children's expectations for story convention.[25, p.36]

5. **Conclusion**

   If alternative 1, the dragon ceremoniously chews and swallows frog (possibly using cutlery and a napkin if the dragon has good table-manners).

   If alternative 2, frog's mouth narrows slowly, dramatically, obviously, and accompanied by a rolling of the eyes and perhaps a froggy expletive (or at least a wailing moan).

6. **Moral**

   None given for this particular case, although young members of the audience can and will supply one (or several) if consulted.

(Adapted from Livo & Rietz, *Storytelling: Process and Practice*[25, pp.35–37])

The following observations can be made when analysing Livo and Rietz' models for storytelling with respect to their applicability to interactive storytelling:

- These models, while useful to a human storyteller, are general in their scope of the themes of stories that can be told. While this increases the themes of the stories that can be told with these models, they may be too general for use in a computerised systems without an extensive knowledge base.

- Some of these models, such as the repetitive system used in *The Wide Mouthed Frog*, are suitable for use in an algorithmic story generator, as repetition of scenes will be a necessity (see Section 6.4.1). However, there is a danger that repetition based stories without significant variation in the scenes may only be of interest to children.

3.3.2 Folktales and Vladimir Propp

If the types of stories to be analysed are too broad, then it is difficult to determine a general system to cope with a wide variety of story types. The ability of a rule based approach to story generation is easier to see in a more restricted domain. This is shown in the work of the Russian folklorist Vladimir Propp. Propp created a classification system for Russian folktales, which he outlines in his book *Morphology of the Folktale*[41]. His method of classification (which separates him from other theorists such as the structuralist Claude Lévi-Strauss) was formalist, meaning the ordering and structure of the plot was the paramount aspect of any folktale. For two folktales to be regarded as similar, the presence of the same characters (such as the witch Baba Yaga, popular in Russian folktales) was not as important as the tales containing the same order and content of plot elements. Propp called these plot elements *functions*, which he defined as 'an act of character, defined from the point of view of its significance for the course of the action.'[41, p.21] Furthermore, Propp stated the following properties of his functions and Russian fairy tales:

1. Functions of characters serve as stable, constant elements in a tale, independent of how or by whom they are fulfilled.

2. The number of functions known to the fairy tale is limited

[41, p.21]

Propp observed that almost all Russian folktales consist of these thirty-one functions[4]. The functions are listed in sequence, and although some of the functions may not be present in a given tale, those that are present are still in their respective order as given by Propp[41, p.22][5]. The folktales could be classified by the functions present in the tale; folktales that contain similar functions are classified together.

The list of Propp's functions is shown in Table 3.1.

[4]The exceptions were tales imported from other regions, such as variants of the Homeric epic poetry, or rare hybrids of these tales. Propp did not think these tales should be properly classified as Russian.

[5]Although not explicity stated by Propp when explaining his sequence, from study of some variants of Russian folktales it seems possible that some disorder or cycles of the functions 12 to 15 (the acquisition of donors and/or magic agents) occur in some tales. However the general principle that Propp intended seems sound.

Vladimir Propp's thirty-one functions of Russian folktales

1. One of the members of a family absents himself from home.
2. An interdiction is addressed to the hero.
3. The interdiction is violated.
4. The villain makes an attempt at reconnaissance.
5. The villain receives information about his victim.
6. The villain attempts to deceive his victim in order to take possession of him or of his belongings.
7. The victim submits to deception and thereby unwittingly helps his enemy.
8. The villain causes harm or injury to a member of a family.

8a. One member of a family either lacks something or desires to have something.

9. Misfortune or lack is made known; the hero is approached with a request or command; he is allowed to go or he is dispatched.
10. The seeker agrees to or decided upon counteraction.
11. The hero leaves home.
12. The hero is tested, interrogated, attacked etc., which prepares the way for his receiving either a magical agent or helper.
13. The hero reacts to the actions of the future donor.
14. The hero acquires the use of a magical agent.
15. The hero is transferred, delivered or led to the whereabouts of an object of search.
16. The hero and the villain join in direct combat.
17. The hero is branded.

18. The villain is defeated.
19. The initial misfortune or lack is liquidated.
20. The hero returns.
21. The hero is pursued.
22. Rescue of the hero from pursuit.
23. The hero, unrecognised, arrives home or in another country.
24. A false hero presents unfounded claims.
25. A difficult task is proposed to the hero.
26. The task is resolved.
27. The hero is recognised.
28. The false hero or villain is exposed.
29. The hero is given a new appearance.
30. The villain is punished.
31. The hero is married and ascends the throne.

Table 3.1: The thirty-one possible functions of Russian folktales, as defined by Vladimir Propp([41], 25–65)

Using this methodology, different story scenes are classified as the same function of fairy tales depending on their contribution to the path of the story, rather than the specific characters As an example, Propp gives the following events from four different Russian fairy tales:

1. A tsar gives an eagle to a hero. The eagle carries the hero away to another kingdom.

2. An old man gives Súčenko a horse. The horse carries Súčenko away to another kingdom.

3. A sorcerer gives Iván a little boat. The boat takes Iván to another kingdom.

4. A princess gives Iván a ring. Young men appearing from out of the ring carry Iván away into another kingdom.

[41, pp.19–20] While the item given, the giver and the hero are different in each instance, the function of each tale fragment is the same. Thus these four scenes would be classified under the same function using Propp's methodology. However, story scenes of different function may contain similar elements. As Propp wrote:

> For example, if Iván marries a tsar's daughter, this is something entirely different than the marriage of a father to a widow with two daughters. A second example: if, in one instance, a hero receives money from his father in the form of 100 roubles and subsequently buys a wise cat with this money, whereas in a second case, the hero is rewarded with a sum of money for an accomplished act of bravery (at which point the tale ends), we have before us two morphologically different elements – in spite of the identical action (the transference of money) in both cases. Thus, identical acts can have different meanings, and vice versa.

So although two story scenes may contain similar elements, such as the granting of a significant sum of money, it is the effect of the act on the path of the story that determines the function of the tale fragment.

While Propp used his functions to determine the similarities between different Russian folktales, researchers into story generation saw the potential of Propp's work in the field; if Propp's functions can be 'reversed' and used as a grammar, they could be used as a set of simple rules to generate folktales. This was the impetus behind the storytelling and creativity system MINSTREL[6] and the main component of a few projects in the area(such as that of Lang's Joseph project[21] and Machado, Paiva and Brna's SAGA[29])

The functions of Propp are closely related to the story mapping presented by Livo and Rietz. The first eleven functions constitute the introduction, and the introduction of the problem. The key problem presented is shown in Function 8, and is often the recovery or restoration of a loved one, or the hero goes off to seek their fortune or a desired object (such as a wondrous magic item, or a bride). Functions 12 to 15 are analogous to events, where the hero gains allies and magic items on their way to face a villain or monster. The resolution of the problem occurs in functions 16 to 19, with a possible conclusion being the hero returning and being rewarded (functions 20 and 31). However, a common variant in Russian folktales is the creation of a new problem upon the solving of the old one (a common example is a false hero claiming the prize that the true hero deserves.) The remaining functions show the new event sequence that accompanies the resolution of this second problem.

While Propp's list of functions is a good starting point for research, a literal implementation is not appropriate for a truly interactive story. There

[6]Scott Turner, the researcher of MINSTREL notes this in his introduction of his thesis, although his actual model has little in common with the work of Propp[62, p.1]

are a number of problems with a strictly Propp based approach:

- Propp's work only deals with Russian folktales, a small domain of stories. Folktales tend to be formulaic, which aids in such a classification. Thus a Propp model would be appropriate only for folktales and closely related kinds of stories.

- A lot of the detail which enriches the folktales, such as specific character traits (for example, the manners and personality of the witch Baba Yaga or the hero Ivan), is not part of Propp's abstractions. This is fine for a classification system, but would need to be implemented in an interactive environment. Some character modelling, such as that used in the projects listed in Section 4.1.1, may solve this; however this has not be implemented in this thesis, and is only partially considered by Propp's classification system (see Section 3.4).

- While Propp can be appropriated for use in automatic story generation (as done by Lang[21]), due to the linear and somewhat fixed nature of the functions for Russian folktales his work is less suitable to an interactive environment for providing agency to the participants without significant adaptations. However, with some changes to Propp's functions to create a set of story functions to adapt them to an interactive domain, this methodology can be used as a basis for an interactive storytelling environment.

While a literal translation of Propp to an interactive domain may prove infeasible, stories can be represented in a fashion that is compatible with Propp's notion of dramatic function. Each 'scene' of a story can be written as a different module. If each scene were to be written as a single module, then this would be similar to a pre-written stories that are available now. Were each possible scene written in this fashion, then this solution would suffer from the exponential growth problem. However, rather than writing each individual scene separately, stories can be represented by a set of dramatic functions. The dramatic functions become the building blocks of the interactive storytelling system.

3.3.3 Myth structure

A more general theory of story structure, which applies to heroic epics and general mythology across the world, was proposed by Joseph Campbell, with his theories on the generalised nature of the structure of the hero myths explained in his work, *The Hero with a Thousand Faces*[4]. His theory extended some of the related studies on archetypes and the collective consciousness proposed by Carl Jung[19]. Campbell identified the similarity of hero tales across different cultures, and postulated that the 'journey of the hero' stories that are a staple in most mythology follow the same patterns and structure. This structure was noticed by story writers, in particular those who write for film. Christopher Vogler, a scriptwriter who is a great adherent of the observations of Campbell, released a condensed version of Campbell's work on story structure called *The Writer's Journey*, which is essentially *The Hero with a Thousand Faces* aimed for scriptwriters. A comparison of the two story structures is provided in Table 3.2.

The Writer's Journey	***The Hero with a Thousand Faces***
Act One	**Departure, Separation**
Ordinary World	World of Common Day
Call to Adventure	Call to Adventure
Refusal of the Call	Refusal of the Call
Meeting with the Mentor	Supernatural Aid
Crossing the First Threshold	Crossing the First Threshold
	Belly of the Whale
Act Two	**Descent, Initiation, Penetration**
Tests, Allies, Enemies	Road of Trials
Approach to the Innermost Cave	
Ordeal	Meeting with the Goddess
	Woman as Temptress
	Atonement with the Father
	Apotheosis
Reward	The Ultimate Boon

The Writer's Journey	***The Hero with a Thousand Faces***
Act Three	**Return**
The Road Back	Refusal of the Return
	The Magic Flight
	Rescue from Within
	Crossing the Threshold
	Return
Resurrection	Master of the Two Worlds
Return with the Elixir	Freedom to Live

Table 3.2: Comparison of outlines and terminology between Joseph Campbell's *The Hero with A Thousand Faces* and Christopher Vogler's *The Writer's Journey*[63, p.12]

Both models follow the 'hero's journey', the path of the hero, as he or she is given a challenge and rises to meet it. Campbell and Vogler noted how each hero's journey goes through a number of different stages, each of which serves its purpose in the developing the story. For example, using Vogler's taxonomy, *'The Ordinary World'* at the beginning of the story sets the scene of the hero's everyday life. *'The Call to Adventure'* is where the hero starts the adventure, and during *'The Refusal of the Call'* the risks the hero must face are outlined. While the specific instantiation will be starkly different from story to story, or film to film (for example, the particulars of *The Lion*

King are different from *Pulp Fiction*), the functionality of the events during each stage with respect to the advancement of the story are similar. Thus this model has been used by scriptwriters as an aid for story development by those writing for the film industry.

These particular models are of interest for two main reasons; firstly they are quite general in that they apply to a wide variety of stories. As Vogler's adaptation of the structure shows, this structure applies to the majority of films produced.[7] Since this structure applies to a hero's journey, it is well suited to use in interactive storytelling. Secondly, since this particular model is well known by script writers, due to the flow of writing talent between the domain of film writing and computer game writing the story structures of Campbell and Vogler are used in the present computer game industry. Due to use of this structure in the design of present computer games, it is practical to consider this model when designing methods for an interactive storytelling system.

However, the major disadvantage of this particular model of story structure stems from its generality. In order for the model to apply to so many different types of hero stories, in each stage the specific definitions of a stage is left open. This general vagueness of the stages makes them more applicable to a wide domain, but makes them less useful as a basis for a computerised model. For example, the 'elixir' mentioned in the stage *'The Return with the Elixir'* is the panacea for the problems plaguing the hero; however this 'elixir'

[7] In *The Writer's Journey*, second edition, Christopher Vogler uses his version of the Hero's Journey to chart the plot of such films as *Titanic*, *The Lion King*, *Pulp Fiction*, *The Full Monty* and *Star Wars*[63] .

can be nearly anything. In traditional adventures, such as the Arthurian legends, the elixir can be a physical object (the quintessential example being the holy grail), whereas in stories about a hero with an behaviour problem the elixir can be the resolution of an internal personality conflict or a change in their state of mind. Since this model relies so heavily on a deep understanding of myth, story and the human condition it is of good use to a human storyteller, however it is more troublesome to use as the basis of a computerised model of stories.

3.4 Archetypes

It is worth commenting on the characters in stories, since realistic characters are one of the objectives that need to be met before good interactive storytelling can be created. Part of character design mentioned in any story model is the notion of *archetype*. An archetype is a constantly appearing character or personality type that features in the myths of all cultures. The theory of archetypes was suggested by Carl Jung, who suggested that these archetypes reflect different aspects of the human mind, stemming from the collective unconsciousness of the human race[19]. These archetypes are a vital part of selecting the 'cast of characters' needed for a story, and are a crucial part of tying together the threads of the story model. There are different variants of what should constitute an archetype, depending on the author and the domain.

Archetypes represent an aspect of humanity or culture, with standard forms of characters used in tales reflecting this. With the children's fairy tales and story structures given by Livo and Rietz[25], good examples can be found in the evil step-mother, or the innocent young child. However, Propp states that like story functions, the archetypes are important for their effect on the story, not for who they are. The prime example of this is the witch Baba Yaga, a recurring character in many Russian fairy tales. While the cannibalistic Baba Yaga is often the antagonist who opposes (and tries to eat) the hero of story, in this role she represents the archetype of *villain*. However, in other stories if the hero is kind to her she will provide assistance to the hero (such as with the gift of a magic item), and in such she is a *helper*.

In Campbell and Vogler's model archetypes also have a strong presence. Important archetypes are the *Hero* (the main character in the adventure), the *Mentor* (who teaches the hero), the *Threshold Guardian* (who tests the hero), the *Herald* (who gives information to the hero and often starts the adventure), the *Shapeshifter* (a character who is not what they first appear), the *Shadow* (another name for villain), and the *Trickster* (who represents mischief and the desire for change)[63]. Like with their model, due to the complexity of the stories that these apply to these concepts are a little more 'fuzzy', with characters often taking more than one role at once (for example, a trainer that is secretly trying to betray the hero would combine the archetypes of Mentor, Shapeshifter and Shadow.

While it is important in a interactive storytelling system to contain a theory of archetypical characters, this concept is not integrated or fully explored in the abstract model covered in this thesis. The nature of archetypes is nevertheless critical to any character modelling in a complete storytelling system, and is an important part of future work required in this area.

3.5 Conclusion

This chapter has covered the nature of stories and discussed the general properties that need to be represented, and the particular models given have been analysed for their suitability to interactive storytelling. It has been shown that a story has a structured nature that is an integral part of its construction. Although the particular structure to be used depends on the genre of the story, any storytelling system needs to represent the appropriate stages or phases of the story type it is designed to tell. The importance of this form means that some plot-level representation needs to be developed as part of the system.

Chapter 4

Existing Methodology and Systems

4.1 Introduction

In this chapter, existing approaches relevant to interactive storytelling are described and analysed for their applicability to the domain. Since a true interactive storytelling system is yet to be developed, the systems described here provide partial solutions or are in related areas that have relevance to this field. Pertinent to this thesis is the research into the area of automatic story generation, a project that devised methods for computer to generate written (non-interactive) stories. Other research approaches try using believable characters to provide an interesting story.

While there seems to be as many ways to write stories as there are authors, most approaches can be divided into one of two schools. The first believe that characters are key to a good story; it is the characters that should be the driving force behind the action in the story. The second believe that the innate structure that drives the plot behind stories is the most important, and the characters should be subservient to the designs of the plot. These two methods are the 'character-based' and 'plot-based' approaches respectively, and are useful to discuss here as they neatly describe not only authoring techniques for stories, but the development work on relevant projects for study when considering interactive storytelling systems. Research falls into either designing better interactive actor agents (the character approach), or into better models of story structure (the plot approach)[11].

4.1.1 Character-based story systems

As evident from the name, it is the characters that are of prime importance in a character-based system. For both traditional media, as well as interactive, the plot of such a story comes purely from the characters interacting in a believable way with each other. The principle is that provided the characters are given interesting personality traits to begin with, and that they are of sufficient depth and complexity, then an interesting story develops[1].

The character-based approach lends itself well to an automated environment. At its most fundamental, this system can be implemented by running a simulation, where the main characters are autonomous computer controlled agents. The level of dramatic experience is then dependant on the depth of the character model present in the autonomous agents in the simulation. These sorts of personality-rich characters are also called *believable agents*

[1] This is an amalgam of the aims of many different character research projects[51, 43, 17, 27, 31, 13]

(taken from the domain of theatre, it refers to characters that allows the audience to suspend disbelief and believe the character is lifelike; for example, cartoon characters like Bugs Bunny are believable if not lifelike[27, 31]). This distinguishes them from realistic agents (which provide realistic behaviour), cognitive agents (which implement models of brain functioning), or rational agents (which aim to act rationally to situations)[45]. The desire for the development of such agents are also helped by their application to many other domains, such as animation techniques, automated actors (working from pre-written scripts) and helper agents or guides[39].

There are a number of research projects that use a character driven approach. Some of these, such as the Oz Project (from Carnegie Mellon)[31], the Software Agents Group (from the MIT Media Lab)[13], and the Virtual Theater Project (from Stanford)[43] are in developing rich believable characters for interactive environments. Others, such as TALE-SPIN[33], and UNIVERSE[23, 24], are story or plot builders that use a character-based approach. Also, due to the applicability to simulations, there are a number of commercial products that employ a character-based agent approach; some closer to the areas of artificial life (such as the *Petz* and *Creatures* series[40, 60]), and some with a more abstract personality representation (such as *The Sims*[32]).

The drawback to a pure character-based system is succinctly shown in Murray's reaction to E. M. Forster's quote (included at the beginning of Section 3.1); with the absence of any higher sense of structure or plot a character-based system resembles a mere chronicle, a sequence of events that only has dramatic potential through the combined fortune and fortitude of the characters' original personalities and design. This is not to say that an interactive character-based simulation is not a worthwhile application; the popularity of the computer game *The Sims*[32] is testament to the profitability of such a product. However a pure character system alone is not a story system. However, the character techniques developed by such systems can (and should) be used in such a system to enrich the resulting stories.

4.1.2 Plot-based story systems

The plot-based story systems are developed around the fundamental idea that there is an intrinsic structure to stories. Stories can be regarded as following a system of rules that is recognised as 'story' structure, similar to how sentences follow the conventions of sentence structure. The sense of what makes a story is acquired by people as part of their development, and is likely to be learnt with language as part of oral language development[25]. Even very young children acquire and have knowledge of (and thus expectations for) the structure of a story [54] (as shown earlier in Section 3.3.1 with the example of *The Wide Mouthed Frog*). This story structure is a product of the development of a culture's perceptions of how the universe works.

If a general grammar (the set of rules) for all stories can be determined then in theory this could be used to generate any story possible. Exactly how formally this structure can be specified is uncertain, as the entire set of possible stories is too complicated to reduce to a simple set of rules (although Rumelhart has produced an approximation of a such a set in order to demonstrate it as a possibility[44]). However there are theorists that have developed sets of story rules that can be used as a starting point for storytelling systems, often by restricting their applicability to a small domain of possible stories. One of those popular with researchers into storytelling systems is afore mentioned Propp, the Russian folklorist who developed a series of rule-like functions to classify Russian folktales[41](described in more detail in Section 3.3.2); while his work was on folktale classification, the grammar-like nature of his classification system has been noted (and applied) by those looking into storytelling systems[62, 6, 29, 16]. Another popular folklorist is Joseph Campbell[4]; while his work is more general in nature (and thus not easily translatable to the computer domain), his model has been used extensively in film and in linear computer narrative development as a basis for story structure[63] (see Section 3.3.3 for more information).

4.1.3 Review of character and plot-based systems

Choosing between the approaches of the character-based or plot-based systems to be the basis of the story generation system may not be strictly exclusive, as a fully implemented interactive story would desirably be built from a hybridisation of both systems. There is no inherent reason why these two elements are incompatible with each other, with a plot system building the story structure and the character system making the cast seem believable. However, this thesis only concerns the plot-based approaches for story generation, for the reason that in the adventurous kinds of stories (the 'quest' tales such as *Beowulf* and *The Odyssey*) that this thesis is considering, it is the plot that is most fundamental. For these kinds of stories the characters, although important to the believability to the story, would have to be secondary to the plot. Secondly, presently there are many character-based approaches stemming from the projects listed above. Some agent based approaches, such as the before mentioned titles *The Sims* and *Creatures*, are commercial successes, and as such there is plenty of worthwhile and current research being completed on believable characters. There is not as much work being completed on the equivalent plot systems today, with most of the work being completed in the 1970s or 1980s[2]. Hence there is more need for research into plot-based methodologies for interactive storytelling.

4.2 Storytelling systems

There have been a number of story generation programs which deal with plot structure and story rules using a variety of different methods. While not being interactive systems, these systems are useful to analyse to assess their potential to be adapted to the interactive domain. Described in this dissertation are some of the more notable representative of story generation systems; while other systems do exist (such as Joseph[21] and SAGA[29]), the systems detailed here provide a good cross-section of the types of systems developed, representing some important milestones in the development of story generation systems, as well as displaying some interesting methods for generating stories.

[2]MINSTREL, one of the later milestones in plot-based story generation, was completed in 1992[62]

4.2.1 Automatic Novel Writer

The Automatic Novel Writer[20] is one of the earliest examples of an automated story writing system. It was developed by Sheldon Klein at the University of Wisconsin in the early 1970s. This particular system is interesting as it outputs relatively long stories for one of the first automated story writing systems. The novel writer (which does not appear to have been given a specific name) outputs short murder mysteries, up to (at least) 2100 words. It is also a good representative of some of the project done at this time by computer scientists researching language generation and linguistics; while including elements of story generation the principle focus is on language generation and the construction of proper sentences.

The plot is created by a simulation of the behaviour of the characters using a series of behaviour rules written by the researchers for each event in the story. Sample of the types of events are 'Two friends meet by chance, they agree to play tennis, one of the friends flirts with the other friend's wife.' and 'Pushing your business partner down the stairs to gain control of the business.'[3] The system also keeps track of time, with various events occurring at different times (such as 'Tea is served at 4 p.m. All guest stop their activities when the butler serves tea'). Some events may provide the triggers for future events (for example, flirting with someone else's spouse provides the trigger for a lover's tryst and adultery, and this tryst provides another trigger, in this case for a murder). The choice of rules is done probabilistically using likelihoods determined by past events and character attributes. Each

[3] Names of the event rules are taken from the comments included within the source provided in the technical report.[20]

character is given a numerical value for a number of personality traits, such as attractiveness, sex drive, intelligence and propensity for violence. Eventually, the novel writer will choose to perform one of the six murder events[4]. This starts the chain of events that starts with the selection of a character to play detective and solve the crime.

```
JAMES WAS VERY RICH.
CLIVE WAS IMPOVERISHED.
CLIVE WANTED THE MONEY.
THE BUTLER WAS RELATED TO JAMES.
THE BUTLER DECIDED TO POISON JAMES.
CLIVE THOUGHT THAT CLIVE INHERITED THE MONEY.
CLIVE KNEW THAT JAMES DRANK A MILK.
CLIVE POISONED THE MILK.
JAMES DRANK THE MILK.
JAMES WENT TO BED.
JAMES DIED.
THE OTHERS THOUGHT THAT JAMES WAS ASLEEP.
CLIVE REMOVED THE FINGERPRINTS.
THE BUTLER RETURNED THE BOTTLE.
```

[4]The six murder events correspond to six different motives, and each have a different method. The six events are: stabbing your spouse for adultery, shooting your spouse's lover, bludgeoning a blackmailer, poisoning a relative for the inheritance, pushing your business partner down the stairs to gain control of the business, and smothering Lady Buxley in a bungled robbery attempt.

```
RONALD AWAKENED.
RONALD GOT UP.
RONALD THOUGHT THAT THE DAY WAS BEAUTIFUL.
RONALD FOUND JAMES.
RONALD SAW THAT JAMES WAS DEAD.
RONALD YELLED.
THE OTHERS AWAKENED.
THE OTHERS RAN TO RONALD.
THE OTHERS SAW JAMES.
EVERYONE TALKED.
HEATHER CALLED THE POLICEMEN.
HUME EXAMINED THE BODY.
DR. BARTHOLOMEW HUME SAID THAT JAMES WAS KILLED BY POISON.
```

Part of a murder story created by the Automatic Novel Writer. The butler, Clive, kills James for the inheritance.[20]

A strong emphasis of the Automatic Novel Writer is on language generation. Much work was put into introducing variation in the sentences provided to make them seem less repetitive. The system also contains some element of character modelling, with the different personality traits corresponding to which roles the characters are given in the story. However, the plot itself is extremely rigid, with little variation in the choice of scenarios offered. Each scenario itself is invariant, with the chain of actions that consist of each scenario pre-written. There is little variation in the stories that the Automatic Novel Writer creates. The pre-written nature of the scenarios in the Automatic Novel Writer makes this model less effective as a basis for an interactive storytelling system, as it would have the same problems as the computer game structures described in Section 2.3.

4.2.2 TALE-SPIN

TALE-SPIN is another early storytelling system, developed by James Meehan at Yale.[33] Using the domain of 'animal stories' (loosely mimicking the style of the fable), TALE-SPIN takes a cast of woodland creatures and modelling the knowledge, plans and goals of these creatures runs through a simulation of how they would achieve their goals.

```
 ONCE UPON A TIME GEORGE ANT LIVED NEAR A PATCH OF GROUND.
THERE WAS A NEST IN AN ASH TREE. WILMA BIRD LIVED IN THE
NEST. THERE WAS SOME WATER IN A RIVER. WILMA KNEW THAT
THE WATER WAS IN THE RIVER. GEORGE KNEW THAT THE WATER
WAS IN THE RIVER. ONE DAY WILMA WAS VERY THIRSTY. WILMA
WANTED TO GET NEAR SOME WATER. WILMA FLEW FROM HER NEST
ACROSS A MEADOW THROUGH A VALLEY TO THE RIVER. WILMA DRANK
THE WATER. WILMA WASN'T THIRSTY ANY MORE.

 GEORGE WAS VERY THIRSTY. GEORGE WANTED TO GET NEAR SOME
WATER. GEORGE WALKED FROM HIS PATCH OF GROUND ACROSS THE
MEADOW THROUGH THE VALLEY TO A RIVER BANK. GEORGE FELL
INTO THE WATER. GEORGE WANTED TO GET NEAR THE VALLEY. GEORGE
COULDN'T GET NEAR THE VALLEY. GEORGE WANTED TO GET NEAR
THE MEADOW. GEORGE COULDN'T GET NEAR THE MEADOW. WILMA
```

```
WANTED GEORGE TO GET NEAR THE MEADOW. WILMA WANTED TO GET
NEAR GEORGE. WILMA GRABBED GEORGE WITH HER CLAW. WILMA
TOOK GEORGE FROM THE RIVER THROUGH THE VALLEY TO THE MEADOW.
GEORGE WAS DEVOTED TO WILMA. GEORGE OWED EVERYTHING TO
WILMA. WILMA LET GO OF GEORGE. GEORGE FELL TO THE MEADOW.
THE END.
```

Sample output from TALE-SPIN.[34, p.91]

The Fox and the Crow

Once upon a time, there was a dishonest fox named Henry who lived in a cave, and a vain and trusting crow named Joe who lived in an elm tree. Joe had gotten a piece of cheese and was holding it in his mouth. One day, Henry walked from his cave, across the meadow to the elm tree. He saw Joe Crow and the cheese and became hungry. He decided that he might get the cheese if Joe Crow spoke, so he told Joe that he liked his singing very much and wanted to hear him sing. Joe was very much pleased with Henry and began to sing. The cheese fell out of his mouth, down to the ground. Henry picked up the cheese and told Joe Crow that he was stupid. Joe was angry, and didn't trust Henry anymore. Henry returned to his cave.

Story generated by TALE-SPIN; reformatted by Meehan for easier reading (story events are left unchanged).[34, p.97]

The above examples show a sample of the stories generated by TALE-SPIN. The initial situation, the environment, the characters and their problems (in this example thirst) are created by the user of the system; the character actions and reactions are automated by the system (in this example Wilma rescues George; there is an automatic motivation for characters to rescue anyone who is in danger of dying). The story is thus generated by a planning system; the characters are given goals, and then try to achieve these goals by the actions available (if none are available, the goal fails).

TALE-SPIN is a purely character driven story building system, and contains no higher level modelling of author processes or plot. As such the stories created read like a log of characters performing simple actions. This means an interactive story system built on the TALE-SPIN model would be similar to those in *The Sims*; a simulation of characters rather than a story structure. This problem was recognised by Meehan; he wrote that the quality of the stories that TALE-SPIN creates depends heavily on the initial settings given by the system user[34, p.96]. In order to get a rendition of a story in the style of a Aesop fable, such as *The Fox and the Crow*, requires careful setup of the environment to ensure the correct result (for example, in one attempt at *The Fox and the Crow* the crow noticed the cheese in its mouth and ate it before the fox ever arrived, making the story pointless. A fix was required to ensure the crow was 'well-fed' at the beginning of the story to stop this occurring[34, p.92]). This kind of care required to the system is acceptable for research, but would not be appropriate for a participant in an interactive story.

4.2.3 MINSTREL

A different approach was taken by Scott Turner when he developed MINSTREL at the University of California.[62] MINSTREL tells stories set in the domain of Arthurian legend, with characters like kings, knights, princesses, hermits, peasants and dragons. Like TALE-SPIN, these characters are modelled with a set of goals and desires, with elements of planning to allow them to achieve their goal. However, there are two fundamental differences from TALE-SPIN:

- MINSTREL includes an *author-level*, a level of control on the story higher than the characters. This author level shapes the story to give it a theme (the 'point' of the story), gives it structure (such as an introduction, a body, and a conclusion), adds in suspense or tragedy where required, and checks for any inconsistencies.[62, pp.24–26].

- MINSTREL uses *case-based* reasoning to choose story events. In other words, it contains a database of story fragments that solve various problems that it can apply or adapt to new events. MINSTREL can modify these story fragments by changing elements of the example, such as swapping a king for a knight, or changing a fatal action to a merely damaging one, to meet the requirements of the story (this was done as model of creative processes)[62, Chapter 3].

Another interesting feature of MINSTREL is in its schema-based approach to story representation. Schemas are based on the work of many artificial intelligence and cognitive scientists, such as Minsky[64] Schank[46, 47, 48], and Dyer[14]. A *schema* is an object-like encapsulation of a concept, such as "John has the goal to satisfy his hunger" (an example taken from MINSTREL[62, p.22]). Each schema has *slot fillers* that when filled with details make the schema as a whole refer to a specific information. Each schema type has specific and limited number of slots, such as for 'authors' and 'objects' in MINSTRELs goal schema (see Figure 4.1[62, p.22].)

An example of the output from MINSTREL, *The Vengeful Princess*, is shown below. Part of the plot has been modified from elements of *Romeo and Juliet*, where the magic potion gives the appearance of being dead now grants transformation into a dragon.

The Vengeful Princess

> Once upon a time there was a Lady of the Court named Jennifer. Jennifer loved a knight named Grunfeld. Grunfeld loved Jennifer.
>
> Jennifer wanted revenge on a lady of the court named Darlene because she had the berries which she picked in the woods and Jennifer wanted to have the berries. Jennifer wanted to scare Darlene. Jennifer wanted a dragon to move towards Darlene so that Darlene believed it would eat her. Jennifer wanted to appear to be a dragon so that a dragon would move towards Darlene. Jennifer drank a magic potion. Jennifer transformed into a dragon. A dragon moved towards Darlene. A dragon was near Darlene.
>
> Grunfeld wanted to impress the king. Grunfeld wanted to move towards the woods. Grunfeld was near the woods. Grunfeld fought a dragon. The dragon died. The dragon was Jennifer. Jennifer wanted to live. Jennifer tried to drink a magic potion

but failed. Grunfeld was filled with grief.

Jennifer was buried in the woods. Grunfeld became a hermit.

Story generated by MINSTREL[62, p.9][5]

Goal Schema

Slots	Meaning
Type	Identifies the particular subtype of goal.
Actor	Identifies the actor who has the goal. For author-level goals, MINSTREL uses a symbol to refer to itself
Object	The story scene or concept to which the goal is applied
Priority	A number from 0–100 indicating the importance of the goal

Figure 4.1: Goal Schema from MINSTREL[62, p.24]

From the perspective of computer writing systems, MINSTREL is a sophisticated storywriter, able to create quite detailed stories when compared to other models. This is mainly due to the author-level refinement that MINSTREL does to improve the story standard. The many different subsystems

[5]The title has been added by Turner, and he also reformatted the output to be more readable (actual output from MINSTREL reads like a telegram, i.e. `GRUNFELD LOVED JENNIFER *STOP*`), but otherwise this is the exact output from MINSTREL.

&GOAL.14

Type:	&Satisfy-Hunger
Actor:	John

Figure 4.2: Example Goal Schema from MINSTREL[62, p.22]

of the author-level, such as the suspense adding system and the inconsistency checker, were added to achieve this. Unfortunately, these subsystems do not convert easily to an interactive domain. The problem lies in that MINSTREL works on a story as a entire entity; it decides on a theme, and determines the main characters and the concluding pivotal scenes, and then searches for suitable motivations to lead to the particular events that lead up to the confrontation, and finally writes the introduction to provide the important personality traits that explain the reasons behind the actions that the characters take in the story[62]. From a non-interactive viewpoint, this methodology is a good way to write a story, and is analogous to how a human writer may write a short story along similar themes as MINSTREL does. However it is not possible to maintain this level of forward planning in an interactive storytelling system which contains a person interacting with the story. Since a person is the interactor who controls the protagonist (the hero of the story), the system cannot predict which choices along the story path the interactor will take. If the interactor is to be given the power of agency, and given the ability to truly influence the path of the story, then an algorithmic story cannot be planned in advance in great detail, because the system must be able to cope with any changes to the plot that will arise from the interactor's actions, and any excess planning may be made irrelevant when the interactor chooses to go a different way from what the system had planned. It is thus infeasible to base the system on a backward planner, like the one used in MINSTREL. Forward planning, where the system does not plan ahead (to any great degree), is preferred to allow interactivity; the interactive storytelling system will have to generate much of the story plot 'on-the-fly', or in other words cannot plan specifically events in the future.

4.3 Conclusion

In this chapter a representative sample of the most significant existing storytelling systems were examined to determine their potential to be adapted for interactivity. The nature of interactivity (discussed in Section 2.2) relevant to that of use in a storytelling system is synonymous with audience choice; an interactive storytelling system must give the audience the ability to make decisions in the path of the story. This differs from automatic story generators such as TALE-SPIN and MINSTREL (see Sections 4.2.2 and 4.2.3 respectively), as their purpose is to write standard linear stories, not to provide audience interaction. Story models that rely on pre-written storylines (which describes the story generation capabilities of the Automatic Novel Writer) are similar to the commercial systems such as computer games described in Section 2.3. Character models (such as TALE-SPIN) have problems with plot centred stories. Furthermore, systems that work on the story as a whole, and plan on changing elements throughout the story as it is written (such a MINSTREL) cannot apply to a story which is being dynamically written to account for the actions of a human participant. So although these systems succeed in their own domain, and may provide some use when designing some of the subsystems of a full system, a method different from all these systems is required for plot generation.

Chapter 5

Theory and Implementation

5.1 Introduction

While the structured nature of stories indicates the feasibility of a computer-based interactive storytelling system, there are many hurdles that need to be overcome. In this chapter, an implementation of the key component of an interactive storytelling system is described and its operability for a full scale system is tested and analysed. This system tests the feasibility of an approach to developing a storytelling model that is capable of responding to the choices made by the interactor of the system.

When considering approaches to interactive storytelling, this thesis has taken a different course from the non-interactive storytelling systems described earlier in the background chapter (provided in Chapter 4). Since the principle of agency is a key focus of this thesis, the argument in this dissertation bases the model for generating interactive situations on an abstraction that is analogous to the interactivity present in the current examples of interactive fiction and computer games. This thesis then shows how this system can be used to model both agency (the choices of the interactor influencing the path of the story), and be extended to match a story structure.

It is beyond the scope of this project to propose and develop a fully operational interactive storytelling system, due to the sheer amount of development and research that still needs to be done in this area. Therefore the ability and techniques to develop a complete interactive storytelling system is not claimed by this thesis. Rather the contribution of this thesis is in the development of a core component of such a system, an abstract model that can be used to implement both agency and storytelling. This abstraction, the 'door and key' model of plot representation, is explained, prototyped and tested in this chapter of the thesis.

5.2 Fundamentals of design

Combining the ideas discussed in previous chapters with a proposed architecture for building an interactive storytelling system, the overview of a possible methodology for interactive storytelling is presented. This generalised structure is suggested as an approach for the design of fully realised storytelling systems.

A complete interactive storytelling system would contain many different subsystems, the exact nature of these depending on the scope and domain of such a system. For example, a complete system may include a character modelling subsystem for believable characters (similar to those developed in the Oz Project[31], the Virtual Theater Project[43], and the other character-based systems described in Section 4.1.1), or analogues of the plot enhancing subsystems included with MINSTREL[62] (although these would have to be adapted for an interactive environment). However, this thesis considers only the core component of plot structure, and the design of an abstraction to be used to develop a sample story system.

Since a storytelling system should emphasise the reliance on structure, then the plot structure itself should be considered as a subsystem, which will ensure that the appropriate story conventions will be maintained throughout the narrative experience that is presented to the interactor. This story structure system is critical as it maintains the highest level of control in the system, and any other story control systems that may be used (such as believable character systems), are superseded by this system. The general principle of the storytelling system is that the path of the story is the most important influence on the interactive experience; other aspects of the story, such as the believability of the characters, are secondary. This is not to deny the importance of having believable characters as an influence on the path of the plot; it is possible and indeed desirable for a believable character subsystem to have some say in the path of the story. The argument of this thesis is that in a plot-based system, if there was a conflict between maintaining a believable character personality and the integrity of the plot, then the plot path takes precedence (in this case, a believable character subsystem could be used to 'justify' any actions a character might take.) However the design of any believable character systems is not part of the scope of this thesis, and is left as further work.

5.2.1 Interactor as protagonist

The central figure in any story is the protagonist, the hero or heroine that is the focus of the adventure. In examples of interactive entertainment, such as in computer games, it is the interactor (the game player) who has control over the protagonist. This distinction between the protagonist and the other supporting characters is assumed in this thesis and in the storytelling system. It is the actions of the protagonist that instantiate the decisions made by the interactor. Thus the protagonist is the most important character in the story, and must be taken into special consideration.

Of course, in a wide medium such as computer games there are examples when there may be more than one interactor controlled protagonist. In strategy and tactical games the interactor is likely to control a team of individuals. The same is true in some role-playing games, such as the *Final Fantasy* or the *Baldur's Gate* series[53, 3], although there is often a prime protagonist that is the centre of the story. Even in a few adventure games there may be many protagonists that can be controlled at the same time (such as *Maniac Mansion* and *Day of the Tentacle*). Some of what is written here may not apply to these multi-protagonist systems. However, the majority of story based games consists of only one hero. As such it is sufficient to limit a storytelling system to only a single protagonist. From now on it is assumed that any story based system referred to is designed for one protagonist only.

The fact that the protagonist is controlled by a human rather than the computer is the key distinguishing factor between a non-interactive storytelling system such as MINSTREL (see Section 4.2.3) and an interactive storytelling system. In a storytelling system that is not interactive, it is possible (even desirable) to plan ahead; it is feasible to build the story in any order, building a story skeleton before filling in the details, or working backwards from the conclusion to fill in the introduction (as done with MINSTREL[62]). However, the presence of a human controlled protagonist does not allow such freedom to manipulate the story structure. The system must work in real time; at any given point in the story the system is unable to change the past (as it has already been presented to the interactor), and the future is unknown due to the unpredictability of the human element. Any system will have to rely on a small amount of planning. If any planning ahead for the remainder of the story is to be done it must be able to cope with the unpredictable nature of the interactor. This is not to say that the system should be totally devoid of planning. The system presented in this thesis uses a mechanism that does not require excessive amounts of planning that ensures compatibility with the requirements of agency.

5.2.2 Scenarios

Since it is not feasible to represent interactive stories using the techniques of pre-written scenes, an algorithmic method of story generation must be used to construct a dynamic story that allows the interactor a sense of agency. A critical part of any algorithmic storytelling system is the methods used for representing the story structure to make it easy to represent on the computer. In this case, consideration of the representation involves the analysis of stories (presented in Chapter 3), to find a pattern that suggests a suitable construction for story fragments, as well as providing the practicality and ease of integrating interactivity to give the interactor the potential of agency within the story.

For this thesis, the system will be modelled on the ideas of Propp and his use of dramatic functions in representing Russian fairy tales (see Section 3.3.2). It is not necessary to provide a faithful adaptation of Propp's functions, as a direct translation of the grammar of functions that Propp developed is not appropriate to the interactive domain. However the idea behind Propp's classification, the definition of the dramatic function, is relevant to the design of the system. To recapitulate on this concept (which was included earlier in Section 3.3.2), the most important aspect of each dramatic function is its purpose in the plot, the effect it has on the story and the hero. This particular aspect determines when the function will be used in the building of the story, and is the prime aspect of the dramatic function. In the system outlined in this thesis, the dramatic functions are called *scenarios* to distinguish them from Propp's functions, as well as to prevent any confusion with the standard definition of 'function' that is present in programming and computer science.

As with Propp's dramatic functions, the more specific aspects of the story, such as the characters, items, and locations involved in each scenario, are not explicitly encoded into the system. These attributes are like slots in the written scenario, waiting to be filled with appropriate choices from those presently available in the story. The story manager that chooses the scenario must instantiate these elements with choices that are appropriate given the present story.

5.2.3 System design

By adopting Propp's approach to fairy tales when designing an interactive storyteller, scenarios (or dramatic function) become the basic building blocks of the system. The goal of the storytelling system is then to combine these blocks in such a way as to provide an interesting story, while receiving input about interactor choices to allow interactivity.

These scenarios represent the fragments of the story that can be built together to form a story. They differ from the rules used in TALE-SPIN and from the schemas used in MINSTREL in that they only apply from the view of the storyteller. Unlike TALE-SPIN and MINSTREL, these scenarios are not used to represent the goals and plans of characters used in the story. The scenarios contain information pertinent only to a storyteller, and to their development of the plot.

An important consideration for each instance of a scenario is the characters contained within. Each scenario generally involves at least one character, and usually two or more. Since these stories are based on the protagonist, and this character is usually the active participant in the scenario, the presence of the protagonist can be assumed in the model. Hence the protagonist does not need to be explicitly encoded into a scenario representation. However many scenarios involve the interaction between the protagonist and another character, such as a donor (a provider of a helpful item) a helper (a character that provides direct assistance to the protagonist), or a villain (the main antagonist in the story)[41, p.79]. Sometimes these characters are created for the singular purpose of fulfilling a role within the scenario (for example, an old man that gives the hero a sword may be present only for this scene), however other times a character will return in subsequent scenes (such as the villain who takes an important role in the shape of the plot, or a helper character who promises to aid the hero later in the story). In this case the system will need to have a mechanism to remember to include such scenes later in the story, while still allowing the interactor to manipulate the flow of the story.

In the present methods used for pre-writing the stories of computer games, the story structure and the scenes within the story are intertwined. Since an interactive storytelling system needs to take an algorithmic approach to story writing, these two elements are best separated. The story structure, the 'big view' of the story, consists of the logical constructs that shape the entire plot path; it is analogous to the functions of Propp and how they link together (as given in Section 3.3.2). The scenes consist the details such as location, characters and items used in each part of the story; the smaller but critically important things that fill out the dramatic content the story. Both of these elements are important to the interactive story.

Creating an equivalent of a scene scriptwriter is very hard. It involves the development of believable characters (such as those described in Section 4.1.1), designing interesting locations and items and how they relate to each other, and a deep knowledge of the craft of writing. The knowledge-base required for this would have to be significant, and the level of craftsmanship the scenes is heavily dependent on the skill of the human author of the system. It is beyond the scope of this thesis to develop such a system, as this is still the fundamental challenge left to the development of interactive storytelling.

What is included in this thesis is an abstract model for the story structure system, a system for interactivity within plots. In the next section, the 'door and key' model is described. In Chapter 6 it is shown that this model can generate an abstraction that maps to the story structures described in Chapter 3. This door and key model can also solve the problem of exponential growth in development time by reusing scenarios within the same story.

5.3 The 'door and key' model

When considering models for interactive storytelling, it is prudent to consider what is currently available and is presently regarded as the best representative examples in this domain to use as the inspiration for the design of a system. With the previously described story creation systems, TALE-SPIN and MINSTREL (see Sections 4.2.2 and 4.2.3 respectively), the modelling was suitable for writing of standard stories, but not that of interactivity. In both the models used by these systems the story scenarios were pre-defined; each story 'action' containing a set of pre-conditions that must be satisfied before the action can be applied, and a set of effects that describe the changes made to the story world when the action is completed.[1] In the standard story writing systems, the system can plan a path of story actions from the starting situation to the desired final situation by finding a chain of actions that perform the required modifications to the story world.

However, this approach, while workable for developing a static story as

[1]These forms of representation are based on the STRIPS planner designed for robot planning[15].

per the aims of TALE-SPIN and MINSTREL, would not work for a dynamic story environment such as one involving a human interactor controlling the story protagonist. Firstly, since the path of the story (including the final situation including the outcomes of the end of the story) depends on the choices made by the interactor as protagonist, as a truly interactive story should be shaped by the interactor to provide a sense of agency. As the interactive is unpredictable in their actions, the system would not know which outcome to aim for. Secondly, due to this unpredictability introduced to the system by the presence of the human controlled protagonist, it is likely that any choice that they make will invalidate any plan that the system has pre-planned for any unexpected behaviour. This can be mitigated somewhat by the use of 'contingency planning', where the system creates a plan for each possible choice that the interactor can make. Unfortunately, a complex story system that provides a strong sense of interactor agency must (by definition) provide a large number of available choices at any single story situation. By offering a large number of choices, planning ahead to a fair depth in the story tree will lead to an unacceptable branching factor for the total set of plans that need to be generated. Any interactive story system that must cope with the element of an unpredictable interactor needs to be able to survive without pre-planning.

For this thesis, the medium that is used as the focus for design is that of computer games, and in particular those that follow the 'adventure game' methodology (see Section 2.1), such as interactive fiction. By basing a system on the interactivity available in these games, it is possible to see how it can be extended to imbuing storytelling with true agency.

As described earlier in the background chapter (see chapter 4), adventure games tend to have linear pre-written stories. However, adventure games need to involve the interactor in some way to distinguish themselves from totally non-interactive media, such as film and books. Thus, in order to maintain the interactor's interest and to provide some challenge to the game, interactive fiction and adventure games link the progression of the story to the involvement of the interactor. As the hero of the story, the protagonist controlled by the interactor, completes certain pivotal actions important to the plot development, then the next step of the story can be revealed, either implicitly (if the actions performed by the hero are story-based) or through a small cutscene where control of the hero is taken from the interactor. These crucial actions are sometimes as simple as the hero walking through a story-critical door, however as this does not involve much ingenuity from the part of the interactor in most adventure games, such doors are often locked. It is up to the interactor to find the key, password, or crowbar to gain entry. Thus the entire interactive nature of such adventure games consists of a series of puzzles, which the interactor has to solve in order to experience the full extent of the pre-written story that forms the backbone of the game[2].

While the standard puzzle-based approach to interactivity used extensively throughout the genre of adventure games can be described as somewhat

[2]This nature of adventure games stories to be dependent on the actions of the interactor was noted by Jonas Heide Smith, who used the apt description of the *interactor as starting gun* to describe this phenomenon so dominant in adventure games. He notes that the puzzle-like nature of adventure games where the interactor has to 'solve' the story in order to experience it might be one explanation as to why so many adventure games use the detective story as their inspiration.[50, p.15]

simplistic (especially as most adventure games only have a single solution to the puzzles that constitute the bulk of their interactive experience), usually the puzzles are tied into the path of the story. If the puzzles were to be automatically generated, then this would be the first step toward granting a sense of agency to the experience of the adventure games, turning them into a truly interactive story[3]. However, in order to automate the content of such puzzles they first need to be generalised into a form that is easily computed. This generalisation can then be used to form a model that can be programmed and analysed.

The *door and key* model is such a generalisation of the standard puzzles that populate the adventure game genre. Under this model the structure of an adventure game is mapped to an equivalent representation consisting of keys and doors. In the equivalent 'door and key' representation of the puzzles, the game is mapped to an environment (called the *story world*) that consists purely of doors and keys. The interactor, in control of the character called the *hero*, can encounter four different constructs which represent the challenges inherent in adventure games: unlocked doors, locked doors, keys and goals.

- *Unlocked doors* represent puzzles, story triggers or any other non-trivial actions that do not require the hero's use of additional items or information located at a different point in the story world. Examples of unlocked doors in adventure games include when the hero enters an important room, or solves a riddle (assuming no external information

[3]The requirement of story generation was described earlier in this dissertation in section 2.3.

is required to find the riddle's solution). Unlocked doors are an important milestone or choice that the interactor can make that splits the 'story world' into two sections.

- *Locked doors* represent puzzles, story milestones or other story critical choices in adventure games that require the hero to do something else before solving it, most usually the retrieval of an item critical to the solution to the problem. A common example of locked doors in adventure games (other than the obvious example of locked doors that require the hero find the key) are characters that agree to help the hero if they give them something that they need. A locked door is linked to its equivalent key.

- *Keys* represent the solutions to the puzzles in adventure games. A key can be anything that the interactor must discover or achieve before solving a 'locked door' puzzle; in adventure games these are usually items to be discovered, but can also be information (such as passwords or treasure maps), or the hero learning a special skill needed to complete a task. A key is linked to a locked door.

- A *goal* is a special construct that is used to represent the completion of the adventure game, and as such there should be only one goal in the story world. The act of obtaining the goal ends the story. It must be emphasised that this goal, like the doors and keys, is an abstract concept that does not represent any specific objective (this is the role of keys and doors) but merely acts as a terminator for the story. As such the presence of the goal in the story space is required as a method to stop the story space being infinite[4]. A goal acts like a key but without the associated locked door.

To summarise: these four constructs are the minimum number required to represent the different types of challenges that are present in interactive story worlds; *Unlocked doors* representing challenges that can be overcome without additional assistance, *locked doors* representing challenges that must be overcome with additional assistance, *keys* representing the additional assistance required for the locked doors, and the special construct the *goal* to act as a terminal to end the story world.

By using the door and key model to act as an abstract representation equivalent to the interactivity in adventure games, the number of effective actions available to the interactor are reduced to a manageable level. Any important actions in the environment are represented by either keys or doors. Under this model, moving between two points in the environment is considered trivial and so is not considered an action that the hero can perform (if moving between two points is non-trivial, due to the presence of an obstacle or a challenge for the interactor, then this should be represented by an unlocked or locked door where appropriate). Under this model, at any one time the hero will be presented with a choice from at least one door, key or goal. The hero can then choose from the following actions to manipulate the story world environment:

- If a *key* is available, the hero can choose to pick it up. The key should

[4]However if an unending story is the desired output, such as that required to model a soap opera (barring the series cancellation), it is possible to model an infinite story space by removing the terminating goal.

be added to the *key list* of the hero, representing the kinds of locked doors that the hero can open.

- If an *unlocked door* is available, the hero can choose to open it. By opening a door new keys, doors or the goal may become available, the equivalent of being able to access the ‘room’ on the other side of the door (see Figure 5.1 for an example).

- If a *locked door* is available, the hero can choose to open it if the appropriate key is in the hero’s *key list* of acquired keys. Like with unlocked doors, opening a locked door will allow access to new keys, doors or the goal.

- Finally, if the hero picks up the *goal*, then the hero has achieved their objective.

5.3.1 Required item lists and key lists

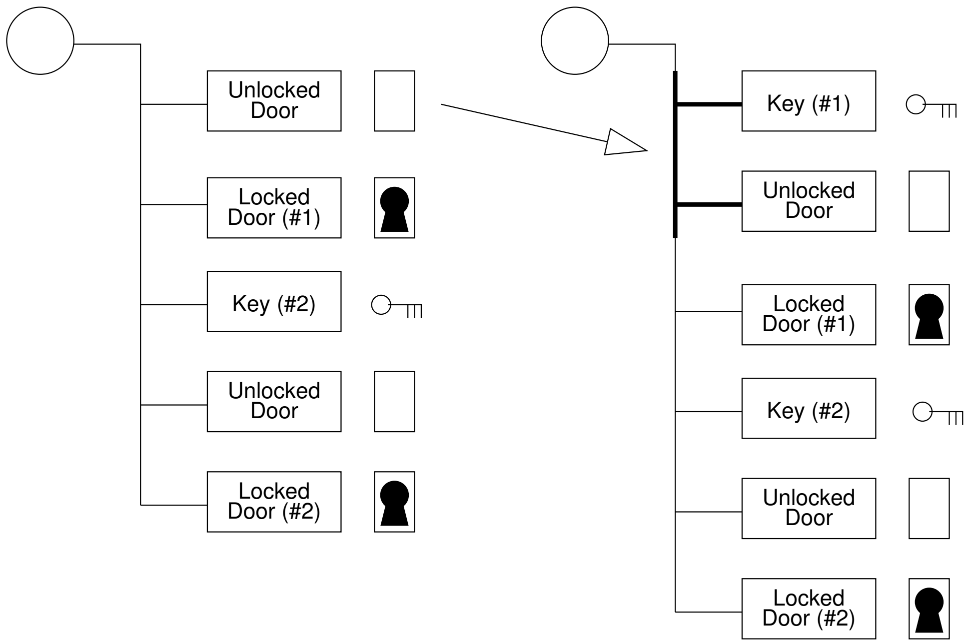
While it has been remarked earlier that planning must be kept to a minimum (to avoid the need for unnecessary changes if the path of the story needs to be changed due to the decisions of the interactor), there needs to be some mechanism for linking locked doors and their keys. The system must remember that for every locked door that is introduced, the respective key must be added at a later stage, and vice versa.

With a backward chaining planner, this can easily be done by introducing the locked doors and keys at the same time. However this approach requires that the story be build in advance, which is counter to the implementation of agency. Since the system to be used will be forward chaining, a different approach is required, however the implementation to generate locked doors and keys is also not difficult. In a forward chaining system, a *required item list* is needed. When a locked door is included into the story world, the key for the door will be added to the list. Similarly, when a key is added to the story world, the locked door is added to the list. The system must then at a later stage include the doors and keys that have been placed in the required item list as elements accessible to the interactor. While simple, this approach leads to the creation of story worlds that are equivalent to that generated by a forward chaining planner, but with the level of planning required to implement agency.

Another construct required by the system is the *key list*. This list records which keys the hero has collected throughout the story; when a key is picked up it is placed on the list. In situations where every locked door and key is unique (i.e. each key opens exactly one door), a key can be removed from the list when it is used. The key list is a representation of which locked doors the hero can enter.

5.4 Conclusion

In this chapter the 'door and key' model with a sketch of the design of a complete interactive storytelling system was presented, and a key component of such a system, an abstract model of interactivity for use in storytelling systems, was given. In the next chapter, an implementation of this model will be outlined. with output showing how this model is applicable to the interactive storytelling domain.



Left side: Before unlocked door is opened

Right side: After door is opened; two new options (key and unlocked door) replace the opened door

Figure 5.1: Effect of opening a door in a 'door and key' story world

Chapter 6

Analysis of the Model

In this section the *door and key* model that was proposed in Section 5.3 is tested, showing how this abstract representation can be used in interactive storytelling. After a short description of the specific implementation of the model that is used, the output is compared with the structure of a typical adventure game format, as well as a Propp based story structure. This section shows how the door and key model is useful from the perspective of the design of adventure games, from the existing interactive stories that are present in the market today, and from that of the traditional story structures, which can be used in future interactive storytelling system design. It also shows how the door and key model satisfies the aim of presenting interactivity at the plot level without needing exponential development time.

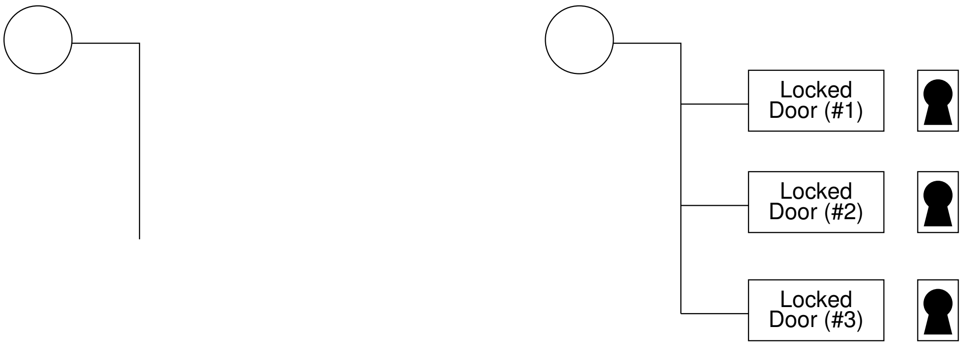
6.1 Building deadlock-free story worlds

The 'door and key' model is a useful representation for the development of interactive stories due to the close correspondence between the existing adventure game structures and that of the simplified story worlds of doors and keys. This model can be used to explore the methods for designing systems for the automatic development of such interactive stories. The first requirement of any such methodology is that any story world created by the system must be *solvable*. A *solvable* story world is one where the hero cannot be trapped in a deadlocked state. In other words it is not possible for the hero to be put in a position in which there is no possible action that can be performed, and it is always possible for the hero to reach the 'goal', or a viable conclusion of the story.[1]

As shown below in Figure 6.1, there are only two different types of states in which the hero cannot perform a valid action. The first is where there are no doors available for the hero to open. Obviously, without any method for progressing the story (which opening a door represents) the story is deadlocked. The second is where the only doors available to the hero are locked with keys that cannot be collected. This scenario is analogous to a badly designed adventure game where the solution to a puzzle needed to clear an

[1]Note that the 'goal' as defined in Section 5.3 is not necessarily synonymous with the success of the hero in the story, but rather represents the story resolution and completion. Depending on the type of story world, the failure of the hero may (or may not) be considered an acceptable resolution of the story. Thus a *solvable* story world does not equate to one that is always successful for the hero, but one where the hero can always progress towards a resolution (even if that resolution is the hero's downfall).

obstacle to the progression of the story is placed beyond the obstacle that needs to be overcome, the equivalent of locking a safe key within the safe itself. Any interactive storytelling system must be designed so a deadlocked state can never be reached.



Left side: No available options for the interactor

Right side: Only locked doors available for the interactor

Figure 6.1: The two states in which a story world can be deadlocked.

The solution to this problem depends on the methodology used by the system. In a backward chaining system, where the story world is generated at once, it is trivial to create locked doors and their keys at the same time; as long as the key is placed on the correct side of the locked door this is an efficient method for creating deadlock-free story worlds. The following simple rules can generate a full door and key deadlock-free story world:

1. Start with the hero having direct access to the goal. In other words the list of available objects the hero can interact with in the starting

state is just the goal.

2. Repeat the following rules as often as needed (the rules are optional, and only apply if the preconditions are met):

   (a) Take an item (door, key or goal) that the hero can access, and put it behind an unlocked door; this unlocked door is now accessible to the hero.

   (b) Take an item (door, key or goal) that the hero can access, and put it behind a locked door; this locked door is now accessible to the hero. The key that unlocks the newly create locked door should also be added to the list of accessible items that the hero can access.

With the backward chaining approach, each element is introduced at a level immediately accessible to the hero. As more elements are added to the story world those elements that are currently accessible may be placed behind newly created doors. With this system any solvable story world can be generated (that is, the hero is guaranteed to be able to reach the goal), and any such story world created will be solvable.

Despite the ease of this backward chaining system, for an interactive storytelling system it has the drawback of completely generating the story world before the hero has a chance to traverse the system. The system will need to be able to construct a deadlock-free story world without too much prior planning, and a different method for building deadlock-free story worlds. The choice of algorithm does not matter from the point of view of the interactor, as all the interactor will perceive is which doors and keys are currently available to the hero. Therefore any system is acceptable as long as it is equivalent to the backward chaining approach listed above.

With a forward chaining system the main complication is with matching locked doors and keys. If a locked door is added to the story world, the system must make sure it adds the appropriate key at a later stage in the simulation. As described in Section 5.3.1, to implement this system a requirements list is necessary to keep track of the paired locked doors and keys to ensure that the story world does not end in deadlock. Whenever a locked door is added to the story world, the equivalent key is added to the requirements list (and vice versa for the adding of keys). The items on the requirements list must then be added into the story world at a later stage in order for the system to work. The only potential problem is if the system is placed in a state where the only doors left are locked with keys that are yet to be added to the story world (providing a state of deadlock). This is not such a problem, as the system can ensure that it always leaves the hero an accessible door that can be opened.

Thus, using this system of story world building, the story world is updated with new keys and doors every time a door is open. An example of a deadlock-free story world generator using this methodology is listed below. This system can generate every possible story world if some of the unlocked doors are considered *'meta-doors'*; intermediatory doors that can serve the purpose of leading to more complicated constructs. For example, to get from one unlocked door to three unlocked doors, the unlocked door can be considered a meta-door that leads to two unlocked doors. One of these two can also be considered a meta-door that leads to two more unlocked doors, making three in total.

6.2 Rules for a deadlock-free story world

In this implementation the story world is initiated with two unlocked doors. Starting with two doors is not mandatory; any feasible story world state could be used as the start state. Technically the most minimalist starting state would require only one unlocked door, however in order to introduce an immediate choice to the interactor two unlocked doors are provided.

When a door is opened, then one of the following sets of new items are added to the list of available doors to the hero. These particular sets have been chosen to cover all possibilities in providing choice to the interactor, with the limiting factor of only introducing a maximum of two new doors, the minimum required to represent an increase in the number of options to the interactor hero. With this minimal set of rules provided all possible story states can be generated. Here the minimal set of rules required is provided with the exception of rules 7 and 8 (one unlocked door and one locked door); these additional rules were included to give emphasis in the creation of locked doors and keys.[2]

It must be emphasised that in order to prevent deadlock, any rule that could possibly lead to a state with no available doors to open (i.e. only locked

[2]Those rules can be discarded with the use of a storytelling heuristic for weighting the choice of rules, or added to if additional emphasis on various story constructions is desired. For example, it may be desired to have the option of replacing a door with nothing to simulate 'dead-ends' or 'red herrings', or simply to cull the possible options to quickly finish for the story.

doors with no keys) may only be applied if there exists another door that is openable by the hero. Thus these rules have the precondition of a further openable door present in the story world that needs to be checked in order to present a deadlock-free story.

The sets of new items are as follows:

1. No doors but a key to a future locked door.

   A check must be done to see if there is another door that the hero is able to open. The locked door that is required is added to requirements list to be added later into the story world.

2. No doors but a key taken from the requirements list.

   Due to the nature of the requirements list the locked door that the key opens will already be available to the hero.

3. One new unlocked door.

   This can model the equivalence of the linear path in interactive stories.

4. One locked door that the key has not yet been generated yet.

   Like before, a check must be done to see if there is another door that the hero is able to open. The required key is added to the requirements list.

5. One locked door taken from the requirements list.

   Due to the nature of the requirements list the key that opens the locked door will already be available to the hero.

6. Two unlocked doors.

   Since this rule replaces one door with two, by considering one or both of the unlocked doors as a meta-door this rule can be used to make any number of items.

7. One unlocked door and one locked door that is linked to a key that has not been generated yet.
   The required key is added to the requirements list.

8. One unlocked door and one locked door that is taken from the requirements list.
   Due to the nature of the requirements list the key that opens the locked door will already be available to the hero.

9. The goal. Needed to terminate the story.

Any combinations not included here, such as an unlocked door and a key, or three unlocked doors, can be created if some unlocked doors are considered meta-doors. For example, two unlocked doors where one door is a meta-door that when opened is replaced with a key, is equivalent to one unlocked door and a key.

6.3 Output of a door and key world generator

The following shows sample output from a Prolog implementation of a door and key world using the methodology listed above. The interactor has the option of opening a door (by typing in the command *open(x)*, where x is the number of the door), or getting a key (*get(x)*, where x is the number of the key), or getting the goal. The output is listed here as it was given from the program, with merely minor formatting changes to distinguish between the input from the interactor and the output from the system.

The doors and keys are all given a unique number, except for locked door and key pairings which share a number between them. Some numbers are missing as the system generates extra doors and keys which are subsequently not used.

In the model below, a *door* is shorthand for an unlocked door. A locked door is shown as *locked_door*. The *current list* is the list of elements that the hero can access or manipulate. The *req. list* is the required item list, and the *key list* contains all unused keys.

```
 Testing door modelling...
Current List:  [door(2), door(1)]
Req.  List:  []
Key List:  []

|:  open(1).
Current List:  [key(3), door(2)]
Req.  List:  [locked_door(3)]
Key List:  []

|:  get(3).
Current List:  [door(2)]
Req.  List:  [locked_door(3)]
Key List:  [3]

|:  open(2).
Current List:  [locked_door(14), door(13)]
Req.  List:  [key(14), locked_door(3)]
```

```
Key List: [3]

|: open(13).
Current List: [key(14), locked_door(14)]
Req. List: [locked_door(3)]
Key List: [3]

|: get(14).
Current List: [locked_door(14)]
Req. List: [locked_door(3)]
Key List: [14, 3]

|: open(14).
Current List: [door(20)]
Req. List: [locked_door(3)]
Key List: [3]

|: open(20).
Current List: [locked_door(29), door(28)]
Req. List: [key(29), locked_door(3)]
Key List: [3]

|: open(28).
Current List: [key(29), locked_door(29)]
Req. List: [locked_door(3)]
Key List: [3]

|: get(29).
Current List: [locked_door(29)]
Req. List: [locked_door(3)]
```

```
Key List:  [29, 3]

|:  open(29).
Current List:  [locked_door(39), door(38)]
Req.  List:  [key(39), locked_door(3)]
Key List:  [3]

|:  open(38).
Current List:  [key(39), locked_door(39)]
Req.  List:  [locked_door(3)]
Key List:  [3]

|:  get(39).
Current List:  [locked_door(39)]
Req.  List:  [locked_door(3)]
Key List:  [39, 3]

|:  open(39).
Current List:  [locked_door(3)]
Req.  List:  []
Key List:  [3]

|:  open(3).
Current List:  [door(44)]
Req.  List:  []
Key List:  []

|:  open(44).
Current List:  [goal]
Req.  List:  []
```

```
Key List:  []

|:  get(goal).
Completed.
```

Figure 6.2: Sample output from Prolog implementation of the door and key model.

6.4 Door and key model using Propp functions

The sample output in Section 6.2 shows the operation of a door and key model generator. This example shows the working of this model and how this model relates to a simple puzzle-based environment[3]. Since an interactive story should consist of structures more complex than merely doors and keys, a comparison between the door and key model and the story structures given earlier in Chapter 3 will be given. For this thesis, the example to be used is an adapted version of the functions of Propp, shown earlier in the dissertation in Table 3.1. The argument for the similarity between these structures can be seen more clearly with an example shown from that of existing adventure games compared with the functions and grammars of Propp (see Section 3.3.2).

Since the formal taxonomy proposed by Propp is inappropriate for direct

[3]Some of the more simplistic action-based games literally only consist of searching for a goal object in an environment containing rooms separated by locked doors and keys

transfer to the domain of interactive storytelling, some adaptation is necessary. For this thesis, a critical aspect is the element of interactivity, so some choice for the interactor is required. Story elements which allow the element of agency must be chosen. Since the options should be significant enough to make a change in the story, for the example provided the decisions are based on two different types of heroes that are prevalent in classic Russian fairy tales.

The first set of options are based on the path of the 'action hero'. This kind of hero prefers to fight, lift rocks, or engage in contests of strength or speed. This type of story features strong warrior heroes reminiscent of the Prince Ivan in Russian fairy tales[1]. In terms of interactive storytelling this kind of character would relate to those people who prefer strong Herculean types of heroes and could easily map to arcade action-based challenges

The second set of options are based on the path of the 'benevolent hero'. This kind of hero helps other people, and receives help in return. This type of story often features more gentle characters, such as fishermen, merchants or young children[1]. In terms of interactive storytelling this kind of character would relate to those people who prefer more helpful, personable heroes and would relate to the conversation puzzles that feature in the classic adventure game genre.

In this example, when the interactor shows a tendency to choose one type of story path over another the system will generate more elements from that kind of story path.[4] Since the system needs to gain some understanding at

[4] The ability to customise a story to better fit the desires of its audience is one potential of a generative interactive storytelling system.

the beginning of the story as to the wishes of the interactor, the first decisions will need to delineate between the two paths. However, there still needs to be some level of agency offered to the interactor throughout the story.

Interactive storytelling using the domain of Russian fairy tales can be provided through a direct linking between the door and key model and equivalent examples from the paths of the 'action hero' and the 'benevolent hero' subdomains of fairy tales. With this model, unlocked doors are mapped to tests that can determine which type of hero the interactor is predisposed towards.

It must be emphasised that this mapping between the door and key model and examples taken from the domain of interactive fairy tales is given here to show the equivalence between the two. It is not provided as a methodology for a fully interactive storytelling system; given here is a one-to-one mapping between door and key symbols and an equivalent in the story domain to show how this model can used in the development of such a system. In a full system, the details of these story scenarios would need to be completely crafted by the computer storytelling system itself, drawing on a detailed knowledge base of rules and information on how to create appropriate storytelling systems. The development of such a system does not fall within the scope of this thesis; the problems that need to be solved in the creation of such a system should be considered unsolved and the crux of the research in providing a true interactive storytelling system. The door and key model presented here is provided as an approach towards the solution of this problem.

Door and Key Symbol	Story Equivalent	Story Example
`door(1)`	Helper hero indicator	going to market
`door(2)`	Action hero indicator	hunting in the forest
`key(3)`	Help a creature	help a bird
`locked_door(3)`	Creature returns favour	bird helps hero
`door(13)`	Go to a new location	travel to a new country
`locked_door(14)`	Monster blocking path	troll guarding bridge
`key(14)`	Clear monster from path	food for troll
`door(20)`	Go to a new location	travel to a new village
`door(28)`	Evaluate hero's type	ask the hero a question
`key(29)`	Get help to go to new location	find a magic boat
`locked_door(29)`	Go to a new location with help	get to a distant island
`door(38)`	Evaluate hero's type	ask the hero a question
`key(39)`	Help a creature	help a fox
`locked_door(39)`	Creature returns favour	fox helps hero
`door(44)`	Box containing item	open chest
`goal`	Treasure	pot of gold

Table 6.1: A sample mapping between doors, keys and Russian fairy tale functions

In this version of the door and key model, the interactions are chosen by choosing the scenario which the hero will solve next. This particular version is, in essence, the same as the door and key scenario that was shown before. However, in this case, the Russian fairy tale equivalent symbols are given instead of the door and key representation.

The next example shown in Figure 6.3 is structurally the same as the listing given in Figure 6.2, however the door and key symbols have been replaced with the equivalent Russian story symbols given in the mapping in Table 6.1. This example shows how the door and key model can be used as a core component of an interactive experience that produces a story similar to that of a Russian fairy tale. To help show the equivalence, the symbols and the choices made by the interactor have been translated into 'story form', a few sentences describing the actions of the hero, Ivan. Scenario generation and selection has not be implemented in the prototype, however the direct mapping between the doors and keys shows how the door and key model can build plots with a developed scene subsystem. The direct mapping also shows how this model can generate stories even if a simple database of scenarios is linked to the system.

```
 Testing story modelling...
Current List: [travel(dark_forest), travel(market)]
Req. List: []
Key List: []

|: travel(market).
Current List: [help(bird), travel(dark_forest)]
Req. List: [get_help_from(bird)]
```

```
Key List: []

|: help(bird).
```

> Ivan goes to the market. While travelling, he meets a bird trapped under a fallen branch, which he sets free. The bird thanks Ivan and promises to help him someday.

```
Current List: [travel(dark_forest)]
Req. List: [get_help_from(bird)]
Key List: [help(bird)]

|: action.
Current List: [block_path(troll), travel(new_country)]
Req. List: [help_clear_path(troll), get_help_from(bird)]
Key List: [help(bird)]

|: travel(new_country).
```

> Ivan enters a dark forest. He reaches a crossroads, where he can travel to a new country or cross a bridge. Ivan chooses the bridge, but it is guarded by a troll, who will only let Ivan pass if he feeds him his favourite goat soup. Ivan travels to the new country.

```
Current List: [help_clear_path(troll), block_path(troll)]
Req. List: [get_help_from(bird)]
Key List: [help(bird)]

|: help_clear_path(troll).
```

```
Current List:  [block_path(troll)]
Req.  List:  [get_help_from(bird)]
Key List:  [help_clear_path(troll), help(bird)]

|:  block_path(troll).
```

There he finds a goat which he makes into soup. Ivan travels back and feeds the troll the soup, and the troll lets him pass.

```
Current List:  [travel(new_village)]
Req.  List:  [get_help_from(bird)]
Key List:  [help(bird)]

|:  travel(new_village).
Current List:  [travel_difficult(island), evaluate_hero]
Req.  List:  [travel_aid(island), get_help_from(bird)]
Key List:  [help(bird)]

|:  evaluate_hero.
```

Ivan arrives at a new village, where he learns of a distant island. At the village an old man tests Ivan with riddles.

```
Current List:  [travel_aid(island), travel_difficult(island)]
Req.  List:  [get_help_from(bird)]
Key List:  [help(bird)]

|:  travel_aid(island).
Current List:  [travel_difficult(island)]
Req.  List:  [get_help_from(bird)]
```

```
Key List:  [travel_aid(island), help(bird)]
```

|: travel_difficult(island).

> Ivan passes the riddles and is given a boat. Ivan sails to the distant island.

```
 Current List:  [get_help_from(fox), evaluate_hero]
Req.  List:  [help(fox), get_help_from(bird)]
Key List:  [help(bird)]
```

|: evaluate_hero.

```
Current List:  [help(fox), get_help_from(fox)]
Req.  List:  [get_help_from(bird)]
Key List:  [help(bird)]
```

|: help(fox).

> There he finds a small hole through the rocks. Another old man tests Ivan with riddles, and points Ivan to a nearby area. Ivan finds a fox in a trap, which he sets free. The fox thanks Ivan and promises to help him.

```
 Current List:  [get_help_from(fox)]
Req.  List:  [get_help_from(bird)]
Key List:  [help(fox), help(bird)]
```

|: get_help_from(fox).

```
Current List:  [get_help_from(bird)]
Req.  List:  []
```

```
Key List:  []

|:  get_help_from(bird).
```

> The fox that Ivan helped goes through the small hole. The bird that Ivan helped flies overhead.

```
 Current List:  [open_chest]
Req.  List:  []
Key List:  []

|:  open_chest.
Current List:  [treasure]
Req.  List:  []
Key List:  []

|:  treasure.
Completed.
```

> The animals spy a chest. Ivan goes to the chest, and opens it. In the chest, Ivan finds enough treasure to make his fortune.

Figure 6.3: Sample output showing Russian fairy tale equivalence of the door and key model shown in Figure 6.2.

The particular example listed in Figure 6.3 shows how this model can be used to provide in an interactive storytelling environment. However, it also shows how dependent the richness of the story world is on the level of detail provided. Without a sentence parser or graphical environment that would be required in the full system, the story will be lacking in detail, as in the example shown above. While the door and key model is a good backbone for a system to provide interactive storytelling, this example shows the ingenuity in encoding the story domain and the effect of the auxiliary display systems. However, it must be noted that although the conversion between the door and key model used above is simplistic, it does show this model can be as the basis for an interactive storytelling system. It also shows how this model can be used to dynamically adapt the story path to facilitate the interactor's sense of agency and story choice.

The ability to adapt to the decisions of the interactor is shown through the particular scenarios the system chooses. The system has full control over which scenarios it chooses at any stage of the story. If the interactor chooses one scenario over another (such as going to the market over exploring the dark forest), or if the interactor makes specific choices within a scenario (such as the responses of the hero when being asked riddles), then the system is free to give scenarios that match the choices of the interactor. While the abstract door and key model presented in this thesis does not implicitly include such scenario choices, its structure is such that it is conducive to such use.

6.4.1 Repetition of used scenarios

By re-using scenarios that have occurred previously, an interactive system can avoid the problem of exponential growth. With only a fixed number of scenarios theoretically an infinite length story can be generated. This is shown in the example in Figure 6.3 with the repeated use of some scenarios, such as `evaluate_hero` and `get_help_from(creature)`.

The following example shows the use of the door and key model with extensive repetition. In this example, the number of scenarios available to the system have been severely limited to show excessive repetition in the output. For this case, all unlocked doors are mapped to moving to a new location, all locked doors are mapped to fighting different types of monstrous creatures, and keys are mapped to specific weapons needed to defeat the monsters represented by the locked doors. One could easily conceive of a system that select corresponding monsters and weapons from a database, to generate a story of sufficient standard for a modern action game.

The pure door and key output is presented in Figure 6.4. This output is used as the framework for the story representation shown in Figure 6.5. In this case, due to the lack of available scenarios the story produced is akin to that present in an action game. A short description of how an action orientated hero would react in this story is presented within the model given in Figure 6.5.

```
 Testing door modelling...
Door List:  [door(2), door(1)]
Req.  List:  []
Key List:  []

|:  open(1).
Door List:  [locked_door(5), door(2)]
Req.  List:  [key(5)]
Key List:  []

|:  open(2).
Door List:  [locked_door(14), door(13), locked_door(5)]
```

```
Req.  List:  [key(14), key(5)]
Key List:  []

|:  open(13).
Door List:  [locked_door(19), door(18), locked_door(14), locked_door(5)]
Req.  List:  [key(19), key(14), key(5)]
Key List:  []

|:  open(18).
Door List:  [key(5), locked_door(19), locked_door(14), locked_door(5)]
Req.  List:  [key(19), key(14)]
Key List:  []

|:  get(5).
Door List:  [locked_door(19), locked_door(14), locked_door(5)]
Req.  List:  [key(19), key(14)]
Key List:  [5]

|:  open(5).
Door List:  [key(14), locked_door(19), locked_door(14)]
Req.  List:  [key(19)]
Key List:  []

|:  get(14).
Door List:  [locked_door(19), locked_door(14)]
Req.  List:  [key(19)]
Key List:  [14]

|:  open(14).
Door List:  [key(19), locked_door(19)]
```

```
Req. List: []
Key List: []

|: get(19).
Door List: [locked_door(19)]
Req. List: []
Key List: [19]

|: open(19).
Door List: [goal]
Req. List: []
Key List: []

|: get(goal).
Completed.
```

Figure 6.4: Sample output from Prolog implementation of the door and key model, used as the framework for the example in Figure 6.5.

```
 Testing door modelling...
Door List:  [travel(new_village), travel(dark_forest)]
Req.  List:  []
Key List:  []

|:  travel(dark_forest).
Door List:  [monster(dragon), travel(new_village)]
Req.  List:  [weapon(sword)]
Key List:  []

|:  travel(new_village).
```

The hero travels into the dark forest.
In the forest, the hero encounters a dragon, but does not have the capability to defeat it.
The hero travels into a new village.

```
 Door List:  [monster(medusa), travel(cave), monster(dragon)]
Req.  List:  [weapon(mirror), weapon(sword)]
Key List:  []

|:  travel(cave).
Door List:  [monster(minotaur), travel(castle), monster(medusa),
monster(dragon)]
Req.  List:  [weapon(axe), weapon(mirror), weapon(sword)]
Key List:  []

|:  travel(castle).
```

In the village, the hero encounters Medusa, but does not have the capability to defeat her.
The hero travels through the village to a cave.
In the cave, the hero encounters a minotaur, but does not have the capability to defeat it.
The hero travels through the cave to a castle.

```
 Door List:  [weapon(sword), monster(minotaur), monster(medusa),
monster(dragon)]
Req.  List:  [weapon(axe), weapon(mirror)]
Key List:  []

|:  weapon(sword).
Door List:  [monster(minotaur), monster(medusa), monster(dragon)]
Req.  List:  [weapon(axe), weapon(mirror)]
Key List:  [sword]

|:  monster(dragon).
```

In the castle, the hero finds a sword.
The hero takes the sword.
The hero defeats the dragon with the sword.

```
 Door List:  [weapon(mirror), monster(minotaur), monster(medusa)]
Req.  List:  [weapon(axe)]
Key List:  []

|:  weapon(mirror).
Door List:  [monster(minotaur), monster(medusa)]
```

```
Req.  List:  [weapon(axe)]
Key List:  [mirror]
```

|: monster(medusa).

The dragon had a mirror.
The hero takes the mirror.
The hero defeats Medusa with the mirror.

```
 Door List:  [weapon(axe), monster(minotaur)]
Req.  List:  []
Key List:  []
```

|: weapon(axe).

```
Door List:  [monster(minotaur)]
Req.  List:  []
Key List:  [axe]
```

|: monster(minotaur).

Medusa had an axe.
The hero takes the axe.
The hero defeats the minotaur with the axe.

```
 Door List:  [treasure]
Req.  List:  []
Key List:  []
```

```
/: treasure.
Completed.
```

The minotaur had the treasure.

The hero takes the treasure.

Figure 6.5: Sample output showing a large amount of repetition of scenarios, using the framework given in Figure 6.4.

As can be seen in Figure 6.5, repeating scenarios can give a story structure, albeit a rather simplistic one. While the action based plot is rudimentary in its content, this particular level of plot detail is representative of that found in these types of games. Note that in an action game the scenarios presented here would represent several minutes of actual game play. There is still potential for agency in games like this, with options between direct combat (fighting a minotaur with an axe) or clever use of tools (such as defeating Medusa with a mirror by reflecting her gaze to turn her into stone). Also note that the structure of the doors in this particular scenario is similar to the cascading problem solving sequence that forms the alternative story event sequence proposed by Livo and Rietz in Section 3.3.1, thus showing the use of another story structure from that of Propp.

The main problem with using repetition is the interactor may become bored through familiarity with the particulars of the scenario. If the audience of a story encounters a similar situation twice, this may break the allure of the story and make the narrative appear robotic and bland. This can be mitigated to a degree with the use of different plot elements to be used in the scenario, such as the use of a bird and a fox in the animal help scenario pairing used in Figure 6.3, or the different types of monsters in Figure 6.5. However, in a full interactive storytelling system more methods will be required in order to mask the repetitive nature of this approach. This could be done by giving the character different personalities, so that no two characters of the same type acted exactly the same. For this approach to work a sophisticated character model, like that used in the Oz Project, Software Agents Group, and the Virtual Theater Project[31, 13, 43] would be required.

6.5 Conclusion

In this chapter, a prototype of the door and key model has been presented. This model has been shown to generate story structures equivalent to those analysed in Chapter 3. The model has also been shown to support the principles of agency through algorithmic story generation. By repeating general scenarios within a single story, the model can avoid the problem of exponential growth in development time which faced pre-written interactive stories. This door and key model is suitable as a core component in a full interactive storytelling system.

Chapter 7

Conclusion

This thesis has discussed the issues involved in developing a fully interactive storytelling software system. It has covered the fundamental aspects of such systems, and explained the nature of the problems involved with the development of interactive storytelling. The thesis has explained the nature of interactivity, and how it can be used to provide agency to the audience of such systems. Storytelling structures proposed by literary theory and structuralists have been provided and discussed with relevance to the problem of interactive storytelling, as well as story creation systems developed by prior researchers for their applicability to the problem of interactive storytelling. From this background material, the general structure and design issues in developing an interactive storytelling system have been discussed, with a general model for interaction, the door and key model, proposed and tested for its applicability to the domain. This general model has been displayed as equivalent to a story structure given by Propp, showing how this could be used as the basis for an interactive method of storytelling. The door and key model presented was shown to be able to manipulate the story to give the interactor a sense of agency, and avoided the problem of exponential development by reuse of generic scenarios. With the support of dynamic scene and character generation this model would lead to a full interactive storytelling system.

To conclude, this thesis has presented a model that can be used as the basis for the building of an interactive storytelling system. While more work is needed in the modelling of interactive storytelling systems, it is worth considering the methods touched upon here when implementing a story within gameplay. The interactive medium, whether through computer games or another method have the potential to be a totally different form of storytelling from the traditional linear methods. Hopefully within a decade the computer can be used not merely as an extension of existing media, but as a fully recognised storytelling medium in its own right.

7.1 Further Development

In such a field as interactive storytelling, where a complete system has yet to be developed, there is considerable scope for further development. One limitation with the door and key model presented in this thesis is in methods for integrating repetition of scenarios in a dramatic interesting fashion (see Section 6.4.1). There are a few different approaches that could be taken to solving this problem.

One such approach is to use repetition to the system's advantage by choosing a story structure that is based on repeating scenarios, such as that used in children's stories like *The Wide Mouthed Frog* (given in Section 3.3.1). This is a simple method of solving this problem as the audience will expect a repetition of story events. However, stories that are built from many repeating scenarios are typically aimed at young children, and so this approach is less useful for stories aimed at a wider audience. Nevertheless, this approach could be used for running substories that thread together within a larger story whole.

Another technique would be to use additional processes to help mask the repetitiveness of scenarios. By implementing a believable character system, the characters within the scenarios will act differently between the otherwise similar situations. Additional subsystems to modify aspects of the scenario, similar to those used in MINSTREL to improve its storytelling[62], can be employed.

One major aspect of interactive storytelling is the development of a scenario generator subsystem, as outlined in Section 5.2.3. This subsystem could conceivably include believable character algorithms, enhanced story world generation, encoding of dramatic themes, computer language systems, and many other areas of research. Many of these areas are already topics of various research groups (such as the character modelling described in Section 4.1.1). However it will be some time before these areas have been significantly developed so they can be combined to create a suitable scenario generator. This is one of the bigger issues for further development into interactive storytelling.

7.1.1 Potential uses of the door and key model in interactive storytelling

The power of the door and key model used in the examples of Figures 6.2 and 6.3 is in its flexibility to adapt to the decisions of the interactor. Since the structure of the story world is being created dynamically by the system it is much easier to shape depending on the interactor's actions. While granting the interactor a sense of agency is a prime consideration, there are further uses for this ability when implemented in a full system.

A further possibility of an interactive storytelling system that can adapt to the desires of the interactor is that the duration of story events or puzzles can be regulated quite easily. Potential scenarios include a player of an adventure game who has spent too long trying to solve a particular puzzle, or a story design that attempts to regulate the duration of each major plot point. Regulation of the timing in the story could also be used to create an episodic structure similar to serials or soap operas; stories where major plot milestones are reached in a timely manner.

Appendix A

Prolog source code

Included here is the Prolog source code for the door and key model example.

```
start :-
    init_world(World),
    write('Testing door modelling...'),
    nl,
    main_loop(World).

% main loop for the demo
main_loop(World) :-
    display_world(World),
    read(Command),
    (Command = stop, ! ;
     do(Command, World, NewWorld),
     main_loop(NewWorld)).
     ).

% displays all important info. in the world state
```

```
display_world((CurrentList, Goals, Keys)) :-
    write('Current List: '),
    write(CurrentList),
        nl,
    write('Req. List: '),
    write(Goals),
        nl,
    write('Key List: '),
    write(Keys),
        nl.

% processing commands
% open a door
do(open(Value), (DoorList, Goals, Keys), (NewDoors, NewGoals, Keys)) :-
    member(door(Value), DoorList),
    remove_door(DoorList, door(Value), DL2),
    choose_new_doors((DL2, Goals, Keys),
        (NewDoors, NewGoals, Keys)).
% open a locked door if the key is in possession
do(open(Value), (DoorList, Goals, Keys),
        (NewDoors, NewGoals, NewKeys)) :-
    member(locked_door(Value), DoorList),
    member(Value, Keys),
    remove_door(DoorList, locked_door(Value), DL2),
    delete(Keys, Value, NewKeys),
    choose_new_doors((DL2, Goals, Keys),
        (NewDoors, NewGoals, Keys)).
% get a key
do(get(Value), (DoorList, Goals, Keys),
        (NewDoors, Goals, [Value|Keys])) :-
```

```
    member(key(Value), DoorList),
    remove_door(DoorList, key(Value), NewDoors).
% signal the wrap up signal, begins the end of the story
do(wrap_up, (DoorList, Goals, Keys),
    (DoorList, [wrap_up|Goals], Keys)).

do(_Command, World, NewWorld) :-
    NewWorld = World, !.

% Initialise the story world
init_world(World) :-
    reset_id,
    build_world(World).

build_world((DoorList, Goals, Keys)) :-
    create_door([], Temp),
    create_door(Temp, DoorList),
    Goals = [],
    Keys = [].

% Create new doors for the story world
choose_new_doors((Doors, Goals, Keys),
        (NewDoors, NewGoals, NewKeys)) :-
    member(wrap_up, Goals),
    findall(World, new_doors_wrap_up(
        (Doors, Goals, Keys), World), WorldList),
    length(WorldList, Length),
    (Length > 0,
        N is random(Length),
        nth0(N, WorldList, (NewDoors, NewGoals, NewKeys)) ;
```

```
    Length = 0,
        NewDoors = [goal|Doors],
        NewGoals = Goals,
        NewKeys = Keys),
    !.
choose_new_doors(OldWorld, NewWorld) :-
    findall(World, new_doors(OldWorld, World), WorldList),
    length(WorldList, Length),
    N is random(Length),
    nth0(N, WorldList, NewWorld).

% add new doors to the world
new_doors(OldWorld, NewWorld) :-
    new_doors_1(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_2(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_3(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_4(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_5(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_6(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_7(OldWorld, NewWorld).
new_doors(OldWorld, NewWorld) :-
    new_doors_8(OldWorld, NewWorld).

new_doors_wrap_up(OldWorld, NewWorld) :-
```

```
    new_doors_0(OldWorld, NewWorld).
new_doors_wrap_up(OldWorld, NewWorld) :-
    new_doors_2(OldWorld, NewWorld).
new_doors_wrap_up(OldWorld, NewWorld) :-
    new_doors_5(OldWorld, NewWorld).

% 0 doors (a special closer rule
%     for use only when there aren't any goals)
new_doors_0((Doors, [wrap_up], Keys), (Doors, [wrap_up], Keys)) :-
    member(door(_), Doors),
    member(locked_door(_), Doors).

% 0 doors - find a (new) key
new_doors_1((Doors, Goals, Keys), ([key(Value)|Doors],
        [locked_door(Value)|Goals], Keys)) :-
    member(door(_), Doors),
    make_key(key(Value)).

% 0 doors - find a (goal) key
new_doors_2((Doors, Goals, Keys),
        ([key(Value)|Doors], NewGoals, Keys)) :-
    (member(door(_), Doors) ; member(locked_door(Value), Doors)),
    member(key(Value), Goals),
    delete(Goals, key(Value), NewGoals).

% 1 door
new_doors_3((OldDoors, Goals, Keys), (NewDoors, Goals, Keys)) :-
    create_door(OldDoors, NewDoors).

% 1 locked door (new)
```

```
new_doors_4((Doors, Goals, Keys),
        ([locked_door(Value)|Doors], [key(Value)|Goals], Keys)) :-
    member(door(_), Doors),
    make_locked_door(locked_door(Value)).

% 1 locked door (goal)
new_doors_5((Doors, Goals, Keys),
        ([locked_door(Value)|Doors], NewGoals, Keys)) :-
    member(locked_door(Value), Goals),
    member(door(_), Doors),
    delete(Goals, locked_door(Value), NewGoals).

% 2 doors
new_doors_6((OldDoors, Goals, Keys), (NewDoors, Goals, Keys)) :-
    create_door(OldDoors, TempDoors),
    create_door(TempDoors, NewDoors).

% 2 doors - one locked
new_doors_7((OldDoors, Goals, Keys),
        ([locked_door(Value)|TempDoors], [key(Value)|Goals], Keys)) :-
    create_door(OldDoors, TempDoors),
    make_locked_door(locked_door(Value)).

% 2 doors - one locked
new_doors_8((Doors, Goals, Keys),
        ([locked_door(Value)|TempDoors], NewGoals, Keys)) :-
    member(locked_door(Value), Goals),
    member(door(_), Doors),
    create_door(Doors, TempDoors),
    delete(Goals, locked_door(Value), NewGoals).
```

```
% add a new 'door' to the current available doors
add_door(Current, Door, [Door|Current]).

% remove a 'door' from the current doors
remove_door(List, Remove, NewList) :-
    delete(List, Remove, NewList).

% make a new unlocked door
make_door(door(Value)) :-
    identifier(Value).

% make a new locked door
make_locked_door(locked_door(Value)) :-
    identifier(Value).

% make a new key
make_key(key(Value)) :-
    identifier(Value).

% create a new door
create_door(OldDoors, NewDoors) :-
    make_door(Door),
    add_door(OldDoors, Door, NewDoors).


% identifier maker; gives a new unique identifier
d_id(0).

identifier(Id) :-
```

```
    clause(d_id(CurrentId), _, Ref),
    Id is CurrentId + 1,
    erase(Ref),
    assert(d_id(Id)).

reset_id :-
    clause(d_id(_), _, Ref),
    erase(Ref),
    assert(d_id(0)).
```

Appendix B

Published Work

The following paper was published (at time of printing) as a result of work done in the thesis:

- David Shaw, Nick Barnes, and Alan Blair. Bringing NPCs to life: adding personality to your computer-controlled characters. In *ADCOG 21: International Conference on Application and Development of Computer Games in the 21st Century*, November 2001.citeShaw:ADCOG

Bibliography

[1] Aleksandr Afanas'ev. *Russian Fairy Tales*. Random House, 1973. translated by Norbert Guterman, illustrations by Alexander Alexeieff.

[2] Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling. *Zork*, 1977.

[3] Bioware. *Baldur's Gate*, 1998.

[4] Joseph Campbell. *The Hero with a Thousand Faces*. Princeton University Press, second edition, 1973.

[5] Lewis Carroll. *Alice's Adventures in Wonerland*. Carroll Foundation, new illustrated edition, 1990. Illustrated by Gavin O'Keefe.

[6] Andy Clarke and Grethe Mitchell. Film and the Development of Interactive Narrative. In Olivier Balet, Grard Subsol, and Patrice Torguet, editors, *Virtual Storytelling : Using Virtual Reality Technologies for Storytelling*, volume 2197 of *Lecture Notes in Computer Science*, pages 81–89. Springer-Verlag, Avignon, France, September 2001.

[7] Lucasarts Entertainment Company. *The Secret of Monkey Island*, 1990.

[8] Lucasarts Entertainment Company. *Grim Fandango*, 1998.

[9] Chris Crawford. Flawed Methods for Interactive Storytelling. *Interactive Entertainment Design*, volume 7, 1993.

[10] Chris Crawford. Fundamentals of Interactivity. *Interactive Entertainment Design*, volume 7, 1993.

[11] Chris Crawford. Assuptions underlying the Erasmatron interactive storytelling engine. In Michael Mateas and Phoebe Sengers, editors, *Proceedings of the AAAI Fall Symposium: Narrative Intelligence*, pages 112–114, November 1999.

[12] Will Crowther and Don Woods. *Colossal Cave Adventure*, 1976.

[13] Trevor Darrell, Pattie Maes, Bruce Blumberg, and Alex P. Pentland. A Novel Environment for Situated Vision and Behavior. In *Workshop for Visual Behaviors at CVPR-94*, pages 68–72, 1994.

[14] Michael G. Dyer. *In Depth Understanding.* MIT Press, Cambridge, Massachusetts, 1983.

[15] R. E. Fikes and N. J. Nilsson. STRIPS: a new approach to the application of theorem proving to problem solving. In *Artificial Intelligence*, volume 2 (3 – 4), pages 189 – 208, 1971.

[16] Dieter Grasbon and Norbert Braun. A Morphological Approach to Interactive Stroytelling. *cast01: Media Art Education*, pages 337–340, 2001.

[17] Nick Hawes. Real-Time Goal-Orientated Behaviour for Computer Game Agents. Technical report, School of Computer Science, The University of Birmingham, November 2000.

[18] id software. *Quake*, 1996.

[19] C. G. Jung. *The Archetypes and the Collective Unconscious.* Routledge & Kegan Paul, Ltd., second edition, 1990. Volume 9, Part I of the collected works of C.G.Jung, Translated by F. C. Hull, Paperback reprint.

[20] Sheldon Klein, John F. Aeschlimann, David F. Balsiger, Steven L. Coverse, Claudine Court, Mark Forster, Robin Lao, John D. Oakley, and Joel Smith. Automatic Novel Writing: A Status Report. Technical Report 186, Computer Sciences Department, University of Wisconsin, July 1973.

[21] R. Raymond Lang. *A Formal Model for Simple Narratives.* PhD thesis, Tulane University, 1997.

[22] Brenda Laurel. *Computers as Theatre*. Addison-Wesley Publishing Company, 1993.

[23] Michael Lebowitz. Creating Characters in a Story-Telling Universe. *Poetics*, 13:171–194, 1984.

[24] Michael Lebowitz. Story Telling as Planning and Learning. *Poetics*, 14:483–502, 1985.

[25] Norma J. Livo and Sandra A. Rietz. *Storytelling: Process and Practice*. Libraries Unlimited, Inc., 1986.

[26] Albert B. Lord. *The Singer of Tales*. Harvard University Press, 1971.

[27] A. Bryan Loyall. *Believable Agents: Building Interactive Personalities*. PhD thesis, Computer Science Department, Carnegie Mellon University, May 1997.

[28] Pascal Luban. Turning a Linear Story into a Game: The Missing Link between Fiction and Interactive Entertainment. *Gamasutra*, June 15 2001.

[29] Isabel Machado, Ana Paiva, and Paul Brna. Real Characters in Virtual Stories: Promoting Interactive Story-Creating Activities. In Olivier Balet, Grard Subsol, and Patrice Torguet, editors, *Virtual Storytelling : Using Virtual Reality Technologies for Storytelling*, volume 2197 of *Lecture Notes in Computer Science*, pages 127–134. Springer-Verlag, Avignon, France, September 2001.

[30] J. M. Mandler. A Code in the Node: The Use of a Story Schema in Retrieval. In *Discourse Processes 1*, 1978.

[31] Michael Mateas. An Oz-Centric Review of Interactive Drama and Believable Agents. In *Lecture Notes in Artificial Intelligence*, volume 1600 of *Lecture Notes in Artificial Intelligence*, pages 297–329. Springer-Verlag, 1999.

[32] Maxis. *The Sims*, 2000.

[33] James R. Meehan. *The Metanovel: Writing Stories by Computer*. PhD thesis, Yale University, 1976.

[34] James R. Meehan. TALE-SPIN, An Interactive Program That Writes Stories. In *Fifth International Joint Conference on Artificial Intelligence*, pages 91–98, MIT, Cambridge, USA, August 1977.

[35] Janet H. Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace.* The Free Press, 1997.

[36] Janet H. Murray. Building coherent plots in interactive fiction. *IEEE Intelligent Systems*, pages 18–21, November/December 1998.

[37] Sierra On-Line. *Kings Quest I: Quest for the Crown*, 1984.

[38] Alexey Pazhitnov. Tetris, 1985.

[39] Paolo Petta and Robert Trappl. Why to Create Personalities for Synthetic Actors. In *Creating Personalities for Synthetic Actors : Towards Autonomous Personality Agents*, volume 1195 of *Lecture Notes in Artificial Intelligence*, pages 1–8. Springer-Verlag, January 1997.

[40] PF.Magic. *Petz series*, 1995.

[41] Vladmir Propp. *Morphology of the Folktale.* American Folklore Society and Indiana University, second edition, 1968. First Edition Translated by Laurence Scott with an Introduction by Svatava Pirkova-Jakobson. Second Edition Revised and Edited with a Preface by Louis A. Wagner, New Introduction by Alan Dundes.

[42] Richard Rouse. *Game Design: theory and practice.* Wordware Publishing, Inc., 2001. author Richard Rouse III, illustrations by Steve Ogden.

[43] Daniel Rousseau and Barbara Hayes-Roth. A Social-Psychological Model for Synthetic Actors. Technical Report KSL 97-07, Knowledge Systems Laboratory, Department of Computer Science, Stanford University, September 1997.

[44] David E. Rumelhart. Notes on a Schema for Stories. In Daniel G. Bobrow and Allan Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 211–236. Academic Press, Inc., 1975.

[45] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Halll, Inc., 1995.

[46] Roger C. Schank. *Scripts, Plans, Goals and Understanding.* Lawrence Erlbaum Associates, Inc., 1977.

[47] Roger C. Schank. *Tell me a story: a new look at real and artificial memory.* Macmillan Publishing Company, 1990.

[48] Roger C. Schank. *The Cognitive Computer: On Language, Learning, and Artificial Intelligence.* Addison-Wesley Publishing Company, 2000.

[49] Johnathan Smith. *Construction Complete? - Computer Gaming's Battle To Take Over The World.* Future Publishing Ltd., 2000.

[50] Jonas Heide Smith. The Road Not Taken. Technical report, Media Studies Graduate Program, The Department of Film and Media Studies, The University of Copenhagen, July 2000.

[51] Sean Smith and Joseph Bates. Towards a Theory of Narrative for Interactive Fiction. Technical Report CMU-CS-89-121, Computer Science Department, Carnegie Mellon University, February 1989.

[52] Paula Underwood Spencer. *Who speaks for wolf: a native American learning story as told to Turtle Woman Singing by her father, Sharp-eyed Hawk.* Tribe of Two Press, 1983. Artwork by Frank Howell.

[53] Squaresoft. *Final Fantasy*, 1987.

[54] N. L. Stein and C. G. Glenn. An Analysis of Story Comprehension in Elementary School Children. In R. O. Freedle, editor, *New Directions in Discourse Processing.* Ablex, 1979.

[55] Andrew Stern. Interactive fiction: the story is just beginning. *IEEE Intelligent Systems*, pages 16–18, November/December 1998.

[56] Laurence Sterne. *The life and opinions of Tristram Shandy, gentleman.* Lehmann, 1948.

[57] Pixar Animation Studios. *Finding Nemo*, 2003.

[58] Nicolas Szilas. Interactive drama on computer: beyond linear narrative. In *AAAI 1999 Fall Symposium on Narrative Intelligence*, 1999.

[59] Nicolas Szilas, Olivier Marty, and Jean-Hugues Réty. Authoring Highly Generative Interactive Drama. In Olivier Balet, Gérard Subsol, and Patrice Torguet, editors, *Lecture Notes in Artificial Intelligence*, volume 2897, pages 37–46, Toulouse, France, November 2003. Springer-Verlag.

[60] Cyberlife Technology. *Petz series*, 1996.

[61] James Tichenor. Story from the Inside Out. *Cyberstage*, 2.1, Fall 2001. http://www.cyberstage.org/archive/cstage21/iastor21.html.

[62] Scott R. Turner. *MINSTREL: A Computer Model of Creativity and Storytelling*. PhD thesis, Computer Science Department, University of California, December 1992.

[63] Christopher Vogler. *The Writer's Journey; Mythic Structure for Storytellers and Screenwriters*. Pan Books, second revised edition, 1998.

[64] Patrick Henry Winston, editor. *The Psychology of Computer Vision*. McGraw-Hill Book Company, 1975.