Story Games and the OPIATE System

Using Case-Based Planning for Structuring Plots with an Expert Story Director Agent and Enacting them in a Socially Simulated Game World.

A thesis submitted to the

University of Dublin, Trinity College

for the degree of

Doctor of Philosophy

Chris R. Fairclough

Department of Computer Science,

University of Dublin,

Trinity College.

October, 2004

Declaration

The work presented in this thesis is, except where otherwise stated, entirely that of the author and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Chris Fairclough

October, 2004.

Permission to Lend or Copy

I hereby agree that the library of the University of Dublin, Trinity College Dublin, has my permission to lend or copy this thesis.

Signed:

Chris Fairclough

October, 2004.

Acknowledgments

Thanks to my project supervisor, Professor Pádraig Cunningham, who has allowed me enormous leeway with my many character flaws, not least of which is my sometimes embarrassingly bad mental arithmetic. This thesis could not have been written without his help and guidance.

Thanks to the all the hard-working souls from the Machine learning group in TCD, Conor who pointed me to Propp and set me on course for the project, Kevin for the extreme guru-ness, and everyone else for being so damn sexy.

Thanks to all my friends some of who still don't quite get what exactly I've been doing all this time, but are still so encouraging.

Thanks Dan for the debugging help and all the moral support.

Thanks to my band, for providing a regular escape from the ivory tower.

Thanks to all my teachers.

Thanks Ste for the help with getting the questionnaire out there.

My family has been a steady source of support for my whole life, and my parents, Chris and Marguerite, are the two most inspiring people I know. Thanks Dad for all your help with this work.

Abstract

Storytelling in computer games has become a major selling point for new titles. With new games integrating compelling storylines with simulated worlds, there is increasingly a standard set of techniques used to tell a story in games, including cut scenes, story-based missions, and the unlocking of new areas of exploration with successful goal completion. To further integrate story telling with the game play of future games, this work draws on storytelling methods in games, along with academic AI techniques and simulation methods, to propose an innovative paradigm for storytelling in future games.

This thesis presents a new approach to creating game mechanics, utilising a number of key concepts that result in an interaction scheme that engages a player with a story, while allowing the player the freedom to interact with and alter that story as it happens. A story director agent was developed that uses case-based planning of skeletal plot scripts, modelled on Propp's morphology, and the dynamic adaptation of these plans.

This agent was incorporated into a social simulation engine that a player interacts with through controlling one of the characters therein. The story director and social simulation are symbiotically linked, with a feedback mechanism that ensures plots are planned consistently with the simulation.

The system was evaluated and analysed and has proved to represent a successful storytelling paradigm, implementing a high level of player interactivity with plotting, while providing an experience that takes the form of an organically whole storyline.

Table of Contents

List of Figures

Related Publications

"Research Directions for AI in Computer Games", Fairclough, C., Fagan, M., MacNamee, B., and Cunningham, P., *in Proceedings of AICS 2001,* pp 333-344., 2001

“An Interactive Story Engine”, Fairclough, C., Cunningham, P., *in Proceedings of AICS 2002, LNAI 2464,* pp 171-176., 2002

“A Multiplayer Case-Based Story Engine”, Fairclough, C., Cunningham, P., in *Proceedings of the 4th International conference on Intelligent games and Simulation. Eurosis*. pp 41-46. 2003

"Story Mechanics as Game Mechanics", Fairclough, C., in *ERCIM News No. 57*, April 2004

"A Multiplayer OPIATE", Fairclough, C., in *the International Journal of Intelligent Games & Simulation Vol. 3 No 2* August 2004

"AI Structuralist Storytelling in Computer Games" Fairclough, C., accepted for publication in *Proceedings of the CGAIDE conference*, Reading UK, November 2004

1. Thesis Introduction

1.1. Introduction

In the early days of computer games, mesmerising flickering screens became portals to areas of adventure and fantasy, as imaginations were ensnared in the first text adventure games such as 'Adventure' by Woods and Crowther , and the 'Zork' series by Lebling and Blank. These games were about exploration of worlds and puzzle solving, told with text from a first-person point of view. Throughout the 1980s, graphics technology was pushed onwards by the ever-increasing demand for more fully realised interactive worlds, and with the advent of Lucasfilm games' acclaimed series of SCUMM games, alongside Sierra's popular King's Quest and Police Quest series , storytelling on computer competed successfully with the 'twitch' arcade games that were becoming all the rage across the western world, yet games had still not lost the taint of being classified as 'kid's stuff', or 'just for geeks'.

Today, games have steadily made their way to a general acceptance as a new medium of entertainment, but with the industry becoming more and more hit-driven, customers don't accept rehashed uninspired games [Laird 2003]. A new, non-licensed high-budget game must have a unique selling point to make money back for the developer, and AI is increasingly touted on the box. Most games nowadays also feature some element of storytelling, and this is another key selling point as games have to compete with the big guns that always make money, such as branded sports games.

As the industry has evolved, a large number of extremely innovative 'game mechanics' have been and gone. The term ‘Game Mechanics’ refers to the set of rules which define the internal workings of the game’s systems and how the player interacts with them. These interaction schemes represent a new medium of expression for those with enough skill to grasp the complexity of systems that create interactive dynamics. This thesis will present a new approach to creating game mechanics, utilising a number of key concepts that result in an interaction scheme that engages a player with a story, while allowing the player the freedom to interact with and alter that story. Stories in commercial games have so far been almost entirely pre-authored, and a small number of academic projects are attempting to create technology for 'interactive storytelling' which automatically tailors a story to a player's interactions. Because academic projects are not at the whim of a consumer market, a large amount of experimentation is permitted in this research, and the techniques used in this work draws from both academic AI disciplines, and storytelling techniques of games.

1.2. Key Concepts

The term ‘interactive storytelling’ is commonly used to refer to the telling of pre-authored stories in an interactive way, but for the purpose of this work another meaning is relevant, that of the telling of stories that are interactive. That is, the story itself is constructed in an interactive way, and then expounded, also in an interactive way. This meaning will be referred to in this work as **istory** (see Figure 1.1). This is in order to avoid confusion, and to emphasise the fact that istory is a distinct phenomenon. The term 'interactive storytelling' emphasises the interactivity of the telling, and thus applies to oral storytelling, pantomime, or puppet-shows, as well as every computer game that has a storyline as motivation for the action. Some oral storytelling is indeed istory telling. Lewis Carroll's 'Alice's Adventures in Wonderland' was originally composed through an ongoing conversation between Carroll and the young Alice Liddell, and it is this concept of a story as conversation that is attempted with the OPIATE system.

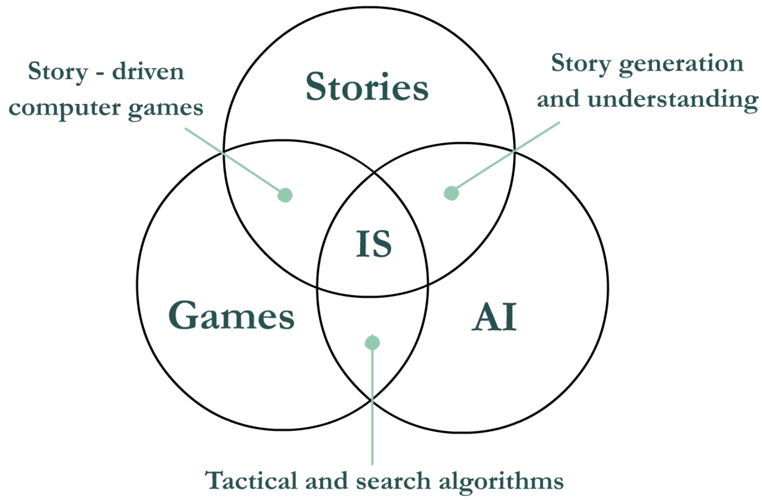


Figure 1.1: Istory as the intersection between Stories, Games, and Artificial Intelligence

To elucidate the acronym **OPIATE –** 'Open-ended Proppian Interactive Adaptive Tale Engine': **Open-ended** - the stories that are told with this system are not forced into a set ending, but allowed to develop based on player choices. Due the plot model used, satisfying resolutions of stories are possible nevertheless. **Proppian** - In 1928, Vladimir Propp published a seminal work in folktale plot analysis that is the basis of this work, and enables the events portrayed by the system to take the form of a story, while being **interactive** and also **adapting** to the actions of the player. The three fields of story analysis, artificial intelligence, and computer games intersect in the creation of true istory where the story is itself interacting with a player. Although the three areas intersect in pairs separately within other areas of study, they converge fully where the story is made as interactive as possible while still retaining the characteristics of a story. This is where the concept that I refer to in this thesis as istory lies.

1.3. Thesis Contributions

- This work developed a new architecture for integrating an accepted AI paradigm, case-based planning, with a storytelling approach based on Vladimir Propp's seminal structural analysis in [Propp 1968].
- The storytelling system adapts to the actions of a player-controlled character within the story world, planning and enacting story plots dynamically. This integrates player agency in three distinct ways.
- The unique storytelling approach is to use a sophisticated AI story director (SD) agent using the case-based planning component in conjunction with an expert system, with the SD controlling the goals of semi-autonomous characters in a socially simulated game world.
- A combination of two general approaches to creating istory is enabled, simulating characters and plot at two different levels, but retaining a symbiotic relationship between the two with feedback loops.
- Through the development of the system, a set of tools was evaluated for use in an author toolkit for creating istory in computer games.

1.4. Thesis Structure

Chapter 2 consists of a discussion of the field of story analysis, as it has been developed from Aristotle of ancient Greece through the dawn of modern scientific enquiry, up to modern analysis such as structuralism.

Chapter 3 explores further background relevant to the development of the system, in three distinct areas. Computer games will be discussed from the point of view of storytelling, and AI techniques that are applicable to istory. Existing Istory projects will also be described.

Chapter 4 discusses the design methods used in creating the OPIATE system, from the design requirements, through the game engine that was developed, to the design of the story telling AI that is the key contribution of this work..

Chapter 5 concerns implementation details of the work, and describes the ongoing testing and refinement of the system, while Chapter 6 contains an analysis of the system and the results of a subjective evaluation that took place.

Chapter 7 finishes the thesis with a summary of key conclusions, a description of possible future work, and a final comment. The References chapter includes a section with all the computer games mentioned in the thesis. The Appendices follow, with some additional information and background on some aspects.

2. Story Analysis

2.1. Introduction

The methodical study of stories and storytelling is an ancient field, but modern technology is allowing the results of these studies to be evaluated in the context of story generation and story telling by computer program. This is generating a new kind of interest in the field of story analysis, and the applicability of the work of past masters is coming under a different type of scrutiny. The structure of this section will follow a chronological path in its discussion of story analysis techniques. Each one of these techniques can lend itself to work in the modern arena of interactive storytelling. Examples of this work in other interactive story technologies will be discussed the next chapter, and compared to OPIATE in the Analysis chapter, Chapter 6 of the thesis.

There will be a more in-depth analysis of the work of Vladimir Propp, as this is the core story analysis methodology used for the development of the system described later. The methodology used in his, and this, work is heavily reliant on the form of story which concerns a hero/villain struggle. This type of story is commonly found in fantasy, comic books, action stories, and also computer games, with an emphasis on the plot as action. This work is meant to facilitate the telling of that type of story in a more interactive way than it is at present, using the medium of computer games; however, it is not meant to directly facilitate a more emotionally focused, character-driven, type of story. The Aristotelian concept of a story as an *imitation of persons acting and doing* is central to this work, rather than a depiction of the characters' emotional states that is found in modern stories, notably novels and movies. Stories can depict emotions in a way that puts the importance of the emotional content of the whole story above the emotions of its individual characters.

2.2. Aristotle and Pre-Modern Analysis

The earliest of the past masters is Aristotle, whose 'Poetics' [Aristotle, 50BC] is a valuable source of analytical tools for modern interactive storytellers. The best known of these tools is the dramatic arc, shown in Figure 2.1, which shows the rise and fall of the audience's tension over the course of the story. There are seven stages of the drama, but the shape of the arc is quite variable.

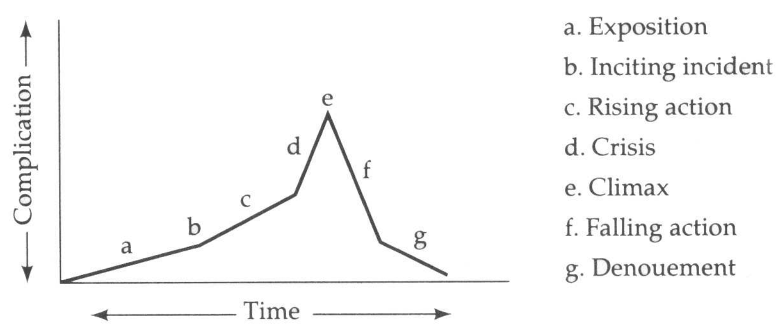


Figure 2.1: The Dramatic Arc

Aristotle also defined six levels at which storytelling takes place. These are, in order of importance: **Plot, Character, Thought, Diction, Song, and Spectacle**. Plot deals with the incidents and the order that they occur. Character principally concerns the moral nature of the agents of the plot. Thought is when 'something is proved to be or not to be' – everything that is asserted. Diction is the words or media that are used to convey the story. Whether any songs are present in a modern story tend to be dependent on genre, yet, where they exist, can serve the purpose of the 'play within a play', or to emphasise a certain incident or character trait in the story, sometimes from a new point of view. This 'song' layer is sometimes referred to as the 'pattern' layer, and this represents a correspondence between layers in their structural features. Thus when a literal song appears in a story it will correspond to themes or plot functions elsewhere in the story. This mechanism can be applied to different levels of the story to pattern a visual event, or a subplot device, around a basic element of the story. Spectacle consists of the visual appearance of the world of the story, emphasising events and components that are visual, whether the set in a play, purely descriptive passages in a novel, or graphics in a computer game.

Aristotle presages many later theorists in his analysis of plot. He introduces concepts such as the importance of causality, the reversal of fortune, and the singularity of the plot. These are important structural features, but the emphasis is on the plot as an organic whole, where every feature is related and a requirement exists that every feature be present only if its absence would upset the integrity of the plot as a whole. Although his work was applied to very distinct genres of play and poetry, dealing with rhythm and meter as well as structural elements, it can still be applied to modern forms of storytelling on computer. One central feature of 'Poetics' is the treatment of a hero's morality. Superior morality is revealed by the hero's action, and must be consistent with the personality of the hero, being either necessary or probable based on previous events. This indicates a fundamental tension between Aristotelian drama-as-action and the interactivity inherent in computer games, where the player character has the choice to act as they wish. An Aristotelian drama is always geared toward the denouement, the elegant conclusion of the plot, which also works in opposition to the freedom of choice of a character in the story. This is the basic problem encountered by those researchers and practitioners currently attempting to present truly interactive computer-mediated stories, what is called in this work istory.

Although 'Poetics' gives us much insight and analytical tools for story analysis, the work of many others has built on this foundation. The variety of storytelling media in modern times means that it is hard to form a cohesive theoretical framework that makes sense across storytelling in all media. The frameworks that make the most sense across all existing media will be the most likely to be useful in new media.

2.2.1. Goethe

During the Renaissance, classical Greek thinkers, through Arabic sources, began to have a profound affect on European culture. Aristotle's concepts of mimesis (from

Latin *imitatio*) and catharsis were popular in literary criticism, yet any new paradigms were, on the whole, unscientific and Romantic. With Déscartes and the onset of the enlightenment, and as the classical view of the world was gradually replaced, a new science began to evolve based on empirical methods. Narratology began to come under the new microscopes, and a number of authors did great work in pushing the field forward. Goethe's enthusiastic analysis of natural phenomena, literary works, and cultural productions in general, was based on a conception of all this from a single point of view. Goethe saw a natural beauty in the best of human productions, and took a morphological approach that unites structures found there with the structures he saw in his biological studies. He identified five types of structural elements in stories (originally in 1797) [Goethe, 1983], and three worlds in which the action takes place:

Structural Elements

1. Progressive elements which promote the progress of the action.
2. Retrogressive elements which divert action away from its goal.
3. Retarding elements which delay or prolong the action.
4. Retrospective elements which incorporate previous events into the action.
5. Prospective elements which anticipate events which are to happen after the action.

Worlds

1. The physical world: the immediate world of the characters, and the natural world surrounding them.
2. The world of the mind: the internal states of the characters.
3. The world of fantasies, premonitions, apparitions, chance, and fate.

Goethe mentions that contemporary writers cannot easily find substitutes for the ancients' miraculous creatures, gods, oracles and soothsayers, to populate the fantasy world. These characters seem to occupy a special place in storytelling, which has preserved them despite their incongruence in a science-drenched culture such as ours. This is due to the deep symbolism that resides in their qualities, which paradoxically has found new means of expression in science-fiction storyworlds. These types of characters are anthropomorphisations of unconscious drives in the human psyche, a revelation that came with the application of scientific methods to the study of the human psyche, by Freud, Jung, and others.

2.3. Empirical Methods

In the 19th century, the anthropologist Adolf Bastian travelled the globe encountering many primitive folk traditions, and developed an idea which he called 'the psychic unity of mankind'. Some of his works from the 1860's have been translated from the German by Koepping and are found in [Koepping, 1983]. Bastian believed that the wellspring of folklore and culture found in the tribes he investigated was the same as that which inspires the development of more complex societies such as his contemporary European world. He analysed the rituals and stories of various South American, Asian and African cultures and found many motifs and ideas that are present around the world, and came to the conclusion that all mankind shares a common psychic make-up that inspires the many and varied artefacts of culture, which becomes apparent in common structural features and devices.

Bastian's work was itself based on previous anthropological work, such as Humboldt's and Herder's, and its foundations were in the gathering of raw data from cultures different from their own, and the forming of a science of man, a 'social physics' [ibid]. Bastian thought that the basis of such a study should be an analysis of popular folk ideas as *patterns of thought as they are acquired in the collective representations of man as social animal*. These folk ideas ranged from the use of the bow and other hunting weapons, to the development of primitive religious concepts. Bastian's approach reflects the increasingly scientific approach being applied to folklore and primitive artefacts of culture. The evolution of ideas and myths from their situated interactions with communities and their environments was a dynamic that was based on Darwin's theories of evolution. Moreover, it is well known that motifs and plots of a lot of literary work, including Shakespeare's plays and Wagner's Operas, have sources in folklore. Folklore by its very nature is not authored, but is formed gradually through human communication by storytelling, ritual and everyday ceremony, and religious practice.

At the end of the 19^{th} century, Veselovskij in Russia was working on categorising Russian folktales according to their motifs – their simplest narrative units [Veselovskij 1913]. Aleksandr Afanas'ev created a comprehensive index of all Russian folktales [Afanas'ev 1976] (originally published 1855-1866). The Formalist school began to build on these, and other works using evolutionist ideas from Germany and England, and Schlovskij noted the irrelevance of studying the genetic origins of the tale without studying the form of the tale [Schlovskij 1965]. The Formalists, however, are thought to have reduced their subjects to pure form, ignoring the content and the origin of the tales they analysed, and the term 'formalist' was originally assigned to the group as a derisory term by contemporary critics such as Bakhtin and Medvedev [Medvedev 1928]. In actuality, the individuals who made up the group took a range of different stances towards the importance of form, and one individual who is often categorised as a formalist was not, although contemporary with it, part of their group. This was Vladimir Propp, who took an approach that abstracted certain forms from the raw material, yet this abstraction was rooted in the specific content of the tales he studied, and emerged from his thorough research of the content of each tale.

2.4. Propp

In 1907, Saussure began an approach to the study of language that was later linked to the inspiration of the structuralist movement, which also begot the field of semiotics [Saussure 1916]. Linguistic structuralism maintained that the meaning of a piece of language was derived from the relationships between the words of a sentence rather than the individual words themselves. At its most basic level, this can be seen in devices such as binary oppositions. For example, the word 'heavy' has no meaning in itself without the concept of 'light'. This is usually noted as the starting point of structuralism, but the application of linguistic devices to the study of literature and stories did not begin straight away. The Formalist group in Moscow in the 1910’s and 20’s was concerned with the rigorous analysis of cultural output using scientific methods. Roman Jakobson, a founding member of the Formalists, was influenced by the work of Saussure, and developed a strain of literary structuralism that was based on the view that the structures behind the work are important, and not the specific subjects of the work [Jakobson 1966]. He provided a link with the general ideas developed by his Russian and Czechoslovakian contemporaries with the Parisian group who later noted the similarity of the approach to Saussure's linguistic methods [Levi-Strauss 1984]. Jakobson was familiar with Propp's work and although it was not translated until later, there was a lot of cross-breeding of paradigms within the European scene, so it is not easy to come to a definite lineage of the ideas behind the bold new theoretical literary movements that emerged at this time.

Vladimir Propp's 'Morphology of the Folktale', published in Russian in 1927 and translated to english in 1968 [Propp 1968], is often seen as merely an introduction to scientific story analysis, yet adaptations and so-called improvements to his work often fail to provide the intuitively satisfying, and scientifically rigorous results of this seminal work. Although Propp was loosely affiliated with the Formalists, his methods did not deal in loose abstractions, as was their norm. They seem to have simplified subjects to the point where many of the intricacies were lost in the analysis, as Eagleton notes in [Eagleton 1996]. He used the term ‘morphology’ to describe his work on the structural analysis of Russian folktales, inspired by Goethe's use of the term in the search for universal organic structures and formations in all life and culture. He treated the material with a rigorous approach that was based entirely on detailed observations of a large corpus of folktales, including different versions of the same tales. This approach was less constraining on the material than the formalists', in my view, as the structures that were imposed on the material were directly derived from the study of the material itself, instead of imposed from outside with only cursory attention to the applicability of the structural analysis to the material.

Propp's treatment centred on the use of the ‘character function’ as the basic building block of the tale. He defined the character function as "an act of character, defined from the point of view of its significance for the course of the action". His astonishing observation was that of the hundreds of tales he analysed, all consisted only of selections from a definitive list of character functions such that, when all were enumerated in the proper order, a single ‘complete’ story would be told. His discovery led him to the conclusion that all plots shared a basic structural template. With a simile using biology, I liken this phenomenon to the similarity of all vertebrates, which all share the same general skeletal template, with some structural features, such as tails or wings, missing, present, or accentuated in specific examples, and an endless variety of different permutations present in nature. The biological metaphor is important, and as Propp disagreed with the formalists on the method of completely abstracting the structures from the material they exist in, he maintained that the skeletal structural entities are lifeless and useless without concrete foundation in the real, fleshed out and given life with the names, motivations, and identities of the characters and their environments.

Propp also worked with the assertion that a character function is independent of the character that performs it. Certain character functions are tied to certain defined roles, yet these roles can be carried out by a number of different characters, and characters can carry out a number of different roles. Assuming that this is true suddenly opens up an avenue of progress for the field of istory telling. The characters and the plot form the two most important parts of a story, and have a symbiotic relationship that must be maintained, yet characters can be interchanged, so long as this does not interfere with the plausibility and suitability of the plot. I would think that this very interchangeability is a trait of stories that helps them endure in human consciousness. This is because of the fact that if an audience can witness a story, and see how the events relate to themselves, or people they know, the story has more powerful resonances with them, thus endures more successfully.

Below is a table which shows the character roles that can be taken by a character in Propp’s morphology (Figure 2.2). The Hero can be classified as a Seeker Hero, or a Victim Hero. The Villain may not be present, and in this case, a 'lack', some requirement of a character, is used as the reason for the Hero's quest, instead of a 'villainy' event. The Mediator character informs the hero of information that is required to further the story. The Donor provides the hero with a magic object for use in the quest, and the Helper helps the hero overcome obstacles in his path. The Princess can fulfil a number of functions but is primarily the hero's final reward, and the FalseHero can be involved near the end of the tale to make false claims about the Hero and the Hero's deeds.

Character Roles	Hero	Villain	Mediator	Donor	Helper	Princess	FalseHero

Figure 2.2: Propp’s Roles

Characters can take one of these roles in carrying out one of the functions from the five groups of functions: Preparation, Complication, Transference, Struggle, and Return. There are 31 specific types of function, each with different subtypes. There will be a detailed description of the different types of function in the Design section of this thesis (section 4.3.1), along with design decisions that were taken to enable the functions to be incorporated into OPIATE.

Propp also developed the powerful concept of a ‘move’ of a story. Although his definitive list of character functions, told from start to finish, form a coherent plot in themselves, a single list can be called a move if it forms a subsection of a longer story. The story as a whole is composed of a number of these moves, and various connective elements and motivations [Propp 1968]. One of the stories detailed in his work concerned two brothers who initially share the same move, then each have their own distinct moves that intertwine in their telling, until at the end one of the brothers defeats the villain who had consumed the other, and rescues him. The moves can be combined in various ways, end to end, intertwined, or parallel. Propp noted that the flexibility of this concept means that his morphology can be applied to many different genres, including mythic forms, and the epic form, where a number of heroes experience their own journeys, alternately aiding each other’s efforts and going on their own quests. If a move features a specific hero, then the other heroes of the story can perform in the other roles the move requires.

Some of Propp’s later works, translated to english and compiled in ‘The Theory and History of Folklore’[Propp 1984], demonstrate his understanding of Bastian's work, yet Propp rejects the idea of psychic unity for a more historical explanation underpinning the similarity of folk concepts around the globe. Propp builds on his own earlier concepts of structure and shows how the evolution of folk ideas, including song, ritual, tales, and dance, is a constant and extremely complex process that results in the proliferation of certain forms and motifs much like how organic creatures evolve. He recognised that myth and folklore share structural elements. As he says, "The study of such ancient works of literature as the Egyptian book of the dead, the myth of Gilgamesh, the myths of ancient Greece, classical tragedy and comedy etc., is indispensable for the folklorist. All this is not folklore, pure and simple; it is reflected and refracted folklore"[ibid]. He also identified similar relationships between folklore and religion, folklore and literature, and folklore and ritual, depicted in Figure 2.3.

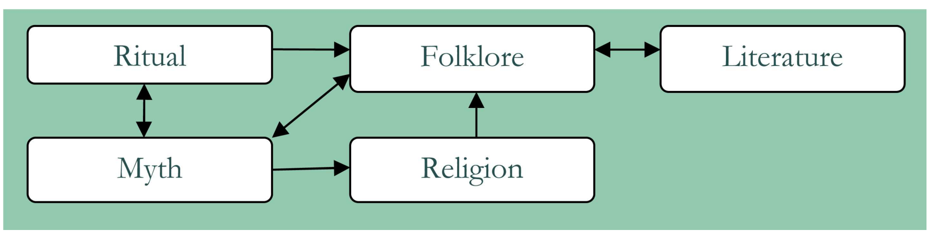


Figure 2.3: Relationships between Types of Cultural Output

The relationship between folklore and literature is more tenuous, as these two forms of expression differ in their usage by lower and higher classes, and tend not to interact in highly hierarchical societies, yet the flow of ideas is not exclusively from folklore to literature, which is the more common. The realm of Folk ideas seems to be the domain of otherwise lost concepts from other areas of human expression, that enable concepts to mutate and cross-breed so that the most hardy and useful will be preserved to occur later, possibly in different forms to their original manifestations. There are certain distinctions, such as the presence of literature on one side of folklore, and ritual and myth on the other. Folklore is intermediate to these. The above discussion is my own paraphrasing of Propp's analysis in [ibid]. There does not seem to be a presence in this exchange for everyday activities, the proliferation of ideas about everyday life and how the folk concerned actually live out their lives. Propp notes that "in the Russian folktale there is not a single credible plot", and although elements of everyday things appear in folktales, they are always background elements. Ritual seems to be the most basic source in this scheme, and Propp notes the similarity of his morphological folktale structure to temporal sequences in ancient initiation and death rituals. The concept of the tale primarily as action and movement is central, as "the journey is the axis of the tale". This is present in stories as a symbol of spiritual growth.

Propp was a firm Marxist, and believed that cultural products were inspired by the people's 'modes of production'. The modes of production of folk art are based on the everyday life of the peasant, and folktales are usually about a conflict or struggle, which is seeded "in the peasants' constant struggle against tyranny of various kinds". Although everyday means of production such as agriculture can be found as motifs in tales, the basic structural template of the Russian tale seems to be more oriented towards hunting rituals and male initiation rituals from ancient times. In the classification of tales, Propp maintains his scientific rigour and agrues that previous classification schemes based on motifs and themes do not prove useful. He argues that because, for example, a motif such as a castle can be found in many distinct types of tale, its use as a classification system, where tales with references to castles would be distinguished from those without, are not useful. He maintains that a classification should be made on structural features using the following criteria:

- The presence or absence of a feature.
- The varieties of a feature.
- Mutually exclusive features.

His classification system, taking the character function as the defining element of the tale, demonstrates remarkable consistency across his corpus of tales, and has been applied to a number of other domains such as Vietnamese, Japanese, and African folktales and various Myths. The fact that Propp has been so influential means that modern stories, told on modern media, can often be composed based on concepts that derive partly from Propp's work, such as the Hero's Journey as conceived by Joseph Campbell [Campbell 1949], and Alan Dundes's work in folklore [Dundes 1969]. Thus, it can be argued that Propp's scheme is not useful for analysis of modern stories, whose authors are aware of his work, and can choose whether to fit the structures of their stories around it or not. The use of Propp's work in the creation of stories is currently just as prevalent as its use in the analysis of them, as it is taught in most folklore studies courses and a number of writing courses [UEA web].

2.5. Structuralism

Claude Levi-Strauss developed his structuralist analysis of Myth, echoing Bastian's assertion that the Myths found in many and diverse native cultures share different structural elements. He defined the 'mytheme' as an elementary unit of myth [Levi-Strauss 1969], yet his use of the term ranges from describing motifs and symbols to structural elements. He did not deal with plot as a temporal structure, but saw the myth as a system of interrelated mythemes. His analysis was not as empirical as Propp's, and he uses abstract concepts, such as binary oppositions, which are reminiscent of formalist concepts. Levi-Strauss noted that Propp's work used many methods that he insists were later developed independently by the French structuralists [Levi-Strauss 1968], and although he admitted that Propp was pre-eminent in the field of folklore studies, he denied that his work has any relevance outside Russian folktales, and cannot be applied to myth and other forms of story.

Algirdas Greimas agreed with Levi-Strauss [Greimas 1966], but Johnathan Culler noted that Levi-Strauss's and Greimas's objection to Propp was that he was 'too empirical' and yet "their own analyses seem to progress without giving any justifications" [Culler 1975]. Previously, Jakobson's analyses had been along similar lines, taking structural, syntactic, and other types of elements and finding parallels between them that he interprets as thematic patterns [Culler 1975]. Even given that these patterns do exist, the practice of grouping elements from different levels of the narrative is not justifiable in a rigorous classification system, and is not useful in finding structural elements common to a wide range of examples. As Propp says, "theme is applicable for a monograph on one plot, but it is inapplicable for scholarly classification" [Propp 1984].

Roland Barthes, one of the most celebrated proponents of structuralism, applies structuralism to narrative with a view to finding abstract frameworks to apply to narratives in general [Barthes 1957]. He used five *codes*, or "voices of which the text is woven":

- Proairetic - concerning the construction of the plot.
- Hermeneutic - the asking and answering of questions, creating suspense.
- Semic - concerning characters.
- Symbolic - the thematic use of symbols.
- Referential – referring to material inside and outside of the text.

These are most reminiscent of Aristotle's levels, yet the inclusion of the referential code reflects the structuralist emphasis on meaning that is endowed by reference instead of innately. Barthes' approach is, in general, deductive. He "proceeds from a hypothetical model of description" to see which narratives "conform to and depart from this model". He recognises the need to work on different linguistic levels, and assumes that the construction of a sentence is analogous to the higher level construction of a plot. Here he acknowledges Propp's work, and admits that Propp uniquely maintained that temporal structure in the plot was important [Barthes 1993].

Structuralism at its most extreme maintains that individual human personalities are not the source of language and literature, but vice versa. A common 'mind' that we cannot directly perceive and of which we are all but a part, directs the development of our ideas, which are not the source of our language, but the result of it. The Czech school developed many of the ideas behind empirical structuralsm that led to this notion [Hajicova et al 1996], but as Levi-Strauss put it, "the mind which does all this thinking is not that of the individual subject: myths think themselves through people, rather than vice versa". Although controversial, and even if it is not grounded in reality, this idea is very compelling to myself as an AI researcher, and it is this concept which may help to allow an istory system to be developed, where an AI agent can structure a plot within an interactive storyworld, taking the role of this 'myth mind'.

2.5.1. Post-Structuralism

Post-structuralism is an umbrella term that refers to the development of a criticism of criticism. Up until and including the structuralist movement, literary criticism was rooted in the comparative analysis of literature itself, and although the academic tradition of acknowledging and upholding previous theoretical works was adhered to, the raw material was the primary subject of enquiry. The structuralist movement gave way to a number of reactionary movements, including deconstruction, pragmatism, hermeneutics, and reception theory. The questions of the older traditions were replaced with new questions that seemed not to enquire about the actual content of the text being studied. Todorov notes [Todorov 1988] that the method of structuralism and the associated field of semiotics were concerned with the question 'what does this mean?'. Deconstruction simply answers 'nothing', pragmatism 'whatever you want it to mean', and hermeneutics and reception theory are more concerned with audience reaction than authorial intent.

It seems to me, although I have necessarily afforded this complex subject limited research, that the modern empirical analysis that has taken place is very much focused on individual works, and progress in classification and useful abstractions is very limited. Modern critics are generally aware of all the methods at their disposal, yet are very individual and subjective in their approaches. Although there is not a proliferation of post-structuralist dogmas in serious philosophy, the deconstructionist ideas of Derrida and his followers still garner a popular following [Eagleton 1996]. I should note here that although my own approach neccessitates the use of formal abstractions in cultural analysis, there are many examples of more subjective methods whose usefulness in their own fields is in no way meant to be diminished by my dimissive treatment here.

Although Carl Jung's discussion of the origin of stories from a psychological point of view is not a direct focus of this work, it must be mentioned here. Jung's *Archetypes* were based on an analysis of hallucinations and dreams from the point of view of unconscious psychological drives. His *Shadow*, *Self*, and *Anima/Animus* are psychological conceptions of the forces at work in our common quest for unity with our own unconscious selves [Jung 1991] (originally published in 1953). The use of storytelling and ritual to aid the quest for self-knowledge that humanity shares, has been fundamental to the development of individuals, and serves a double purpose in its ability to elucidate ethical and moral themes for new generations. Jung argued that the character of a 'damsel in distress' represented the Anima - the view of the unconscious self, and the Shadow as those forces in an individual's psyche which prevent the atonement of an individual with their own unconscious. Northrop Frye viewed mythologies as an expression of this deep struggle, and conceptualised the tension between science and mythology as a conflict between a society's freedom and society's concern [Frye 1957], thus widening the percieved usefulness of storytelling in modern society.

2.6. Stories in Chess

The game of chess has been mastered by IBM's Deep Blue, using methods which evaluate up to 200 million positions per second [IBM web]. Newer attempts to create AI chess players incorporate models of strategic thinking, with positional analysis and opponent modelling to enable non-brute force methods, and there are also more *believable* chess opponents that are programmed to make mistakes sometimes [Fritz web]. This does not seem to be relevant to our discussion on story modelling, but the game of chess is interesting if you look at each game as a self-contained story. Lewis Carroll admitted that the events in the sequel to ‘Alice’s Adventures in Wonderland’, 'Through the Looking glass and what Alice found there' in fact represented a chess game in which all the character activities were analogous to a set of valid chess moves [Gardner 1960]. Carroll’s abilities as a logician enabled him to play with the causal and logical intricacies of the story world to fulfil a number of artistic objectives simultaneously.

The applicability of story analysis to the game of chess is not very deep. However, it is relevant and provides an introduction to the next chapter, which deals with computer games in section 3.2. Each chess piece can be seen as a character, with each type of piece being a role. There are a large number of pawns, and these can be seen as playing a relatively unimportant role, protecting other pieces and creating inroads. Rooks, knights and bishops play a more strategic role in constructing attacks, and the queen and king can be seen as the main protagonists, as well as the main villains. Characters can appear to make choices about whether to be defensive, protect a certain other character, or sacrifice themselves to obtain a better position for the team. Taking each move at time, I analysed a 'fool's mate' game of chess with a view to interpreting the moves as valid character actions in a story's plot. I assumed that taking such an elementary game, in which two beginners might participate, would result in a somewhat simple story. Seven separate pieces move in this game, each assuming a role in the story. The table below in Figure 2.4 shows the chess moves and corresponding story events I have interpreted from them, and Figure 2.5 shows the pieces on the board just before the end.

Chess Moves	**Story Events**
WHITE: KINGPAWN TO D4	A black bird flies out to spy on a small house in the woods.
BLACK: KINGPAWN TO D5	A small boy ventures out from the house to look at the bird in a clearing.
WHITE: QUEENPAWN TO E3	The evil woodsman orders a wolf to follow the bird and kidnap the boy.
BLACK: QUEEN TO C6	The boy's mother leaves the house to find out where the boy has gone.
WHITE: KNIGHT TO F3	The woodsman's henchman, a dark rider, closes on the clearing from the side.

BLACK: BISHOP TO F5	The boy's big brother leaves the house after his mother and notices the rider.
WHITE: BISHOP TO D2	The woodsman hides behind a tree.
BLACK: QUEEN TAKES C2 CHECKMATE	The mother sees another wolf and uses her magic to make the wolf eat the woodsman, protected by the big brother.

Figure 2.4: Chess Moves interpreted as story events, with each character represented by a chess piece

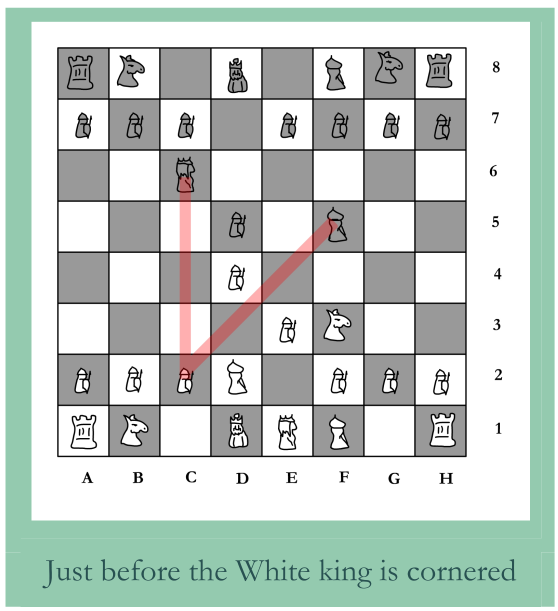


Figure 2.5: The Chessboard

What can be seen from this example is that although the pieces are commonly seen as mere markers in a game, they can readily be anthropomorphised. The game is a complex system, and because the pieces are assigned names that facilitate the process, we can ascribe character intent to them. I have decided to use the game of chess as a way of demonstrating that game dynamics can be interpreted as a story. The rule set of chess and the chess pieces also are related to the story schemes shown earlier, particularly Propp's scheme, where each role is limited to its own discrete set of functions in the overall scheme.

There have been studies [Koepping 1983] that show how myths in native cultures are used to represent natural and systemic forces as characters with intents. The first step in this process is trying to envisage the system from the point of view of the component object, such as the chess piece, so that a subjective view of the system is generated which informs the anthropomorphisation of that object. Although the ability to recount a chess game as a story does not mean that chess is storytelling, the system that is chess is constructed in a way that actively encourages the process of interpretation as a story. Some computer games, such as 'The Sims' by Maxis, have also been constructed like this. The Sims and its creator's work will be discussed in section 3.2.5.

2.7. Conclusions

This chapter has introduced some esoteric (to the average computer scientist) areas of study, and thus this section provides an elaboration of why all of this material is relevant to the creation of an istory system. Because the OPIATE system's purpose is to allow the creation of coherent and believable stories in real time with a human interactor, a more scientific understanding of the workings of stories as a component of human culture is desirable, as a means to abstracting some of these workings for implementation as a computer program.

Aristotle's analyses were seminal and remain the bedrock for much of the work in the field, so it was necessary to investigate how his view of plotting and organic wholeness of a story could be integrated with an istory system. The layered approach was instrumental to implementation. The morphological approach of Goethe profoundly influenced Propp and other structuralists' work, so it had to be mentioned, and his 5 elements and 3 worlds model (section 2.2.1) is useful as it clarifies the later work as well as being a useful categorization technique for segments from Propp's morphology. Although Propp uses many more elements, each of them could be classified into one of

Goethe's five types. This was not explicitly modelled, but helped in the implementation of the system. I thought that an elaboration of Propp's origins and contemporaries was necessary, because the uniqueness of his approach and results is sometimes ignored, and I have found many references in the literature which indicate that Propp's work has since been improved, and while this may be so for the purposes of literary analysis, for my work in real-time plot generation, Propp's schema was the most useful.

Structuralism's 'myth mind' concept, along with Jung's collective unconsciousness, are both early scientific research into a concept that I believe is most important in AI research in general, and especially in implementing an AI that attempts to perform some activity normally seen as creative. This concept is that of intelligence as an artefact of the workings of a system, applied to systems made up of people. Just as ants exhibit intelligence at the hive-mind level, and as individual brain cells are not aware of the intelligence they exhibit as a group, the idea of de-centralised systems that exhibit AI is related to the structuralist view that the structures of language have a defining effect on concept-forming in individual human beings. Whether there is a human-recognisable consciousness at this higher level is not important for this research, but the fact that there is a certain amount of structure and pattern at this higher level of the system – i.e. the structure of languages themselves - which influences the evolution of the system, means that the concept could be used in AI research to influence the evolution of a population by varying the communication structures that are available to them.

As will be seen later, the OPIATE system relies on a higher-level 'omniscient' Story Director AI agent which controls lower-level characters in a way that aims to portray a story to the user. Further application of the myth-mind and collective unconsciousness concepts to AI development relates to future work postulated in chapter 7, specifically section 7.4.6, yet for the most part is outside the scope of this thesis and is mentioned here because it is a natural extension of the OPIATE system. Also, Jung's work is mentioned because his Archetypes correspond roughly to the character roles in Propp's morphology, which is an argument for the inclusiveness of Propp's scheme, and it's applicability to different genres of story.

Finally, in section 2.6 of this chapter that deals with chess, I aim to show how stories are not limited to static representations, but can be encoded in the definition of a game system. The naming of pieces, and the differentiation in the roles of pieces, encourages the interpretation of a game system's dynamics as a story. This view is my own, supported by my analysis of computer games in the next chapter.

3. Modern Background Work

3.1. Introduction

This chapter is a review of some further background that is relevant to this thesis. The elaboration of the thesis requires some background in three distinct fields that have exhibited some overlap as the understanding of stories within an interactive framework has developed. These fields are **Story Analysis**, **Computer Games**, and **Artificial Intelligence**, AI. Story analysis has been dealt with in the previous chapter, and this chapter has sections dealing with games (section 3.2) and AI (section 3.3). These precede a section describing how the three fields intersect in the development of interactive story systems (section 3.4), with some justifications and general conclusions in section 3.5.

3.2. Computer Games

The computer games industry is an environment where the general practices, algorithms, and approaches in use have grown based on the evolutionary pressures of the market. There has been remarkable diversity in the products that have been developed, but certain game genres have emerged. All of these genres can now incorporate the framework of a story to enable more player involvement in the game. The standard approach is to introduce the main character and establish some other characters, and then introduce either a villain or some task that must be accomplished. The player is then put in control and must overcome the villain or the task. The story elements are usually conveyed by either using cinematic 'cut scenes' that are not interactive, or by limiting the possible options of the player so that these events must take place within the game environment.

3.2.1. Action Games

No single method for portraying a story has been found to be the most successful, and one popular genre is almost bereft of one. This genre, the first person shooter (FPS) draws on a peculiarly masculine instinct to destroy everything in sight. What stories there are in these games are necessarily simplistic, because the character interactions largely involve guns. There are a few exceptions, and these tend to be very successful titles. ‘GTA Vice City’, ‘Half-Life’ and ‘Mafia’ all represent different approaches to telling an involving story in an action game environment, where virtual gun-toting is the primary means of interaction with the game.

GTA: Vice City by Rockstar North	Half-Life by Valve	Mafia by Illusion Softworks

Figure 3.1: Action-based story games

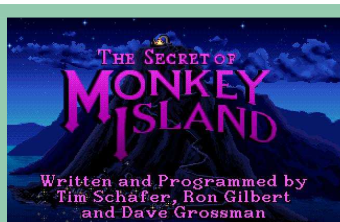
It has been argued [Aarseth 1997] that games and stories are fundamentally incompatible. Games such as the three mentioned above show this to be fallacy. Aarseth views the narrative and game components of such software titles as divorced from one another, as separate components, but the most successful examples of the phenomenon are to be found when the gameplaying and storytelling are integral, and integrated, parts of the same experience. For example, in Half-Life, the content of the story itself is not terribly original (humans inadvertently facilitate alien invasion, lone hero fights them back), but the story is carefully told exclusively through the first-person perspective and by giving the player parts of the storyline bit-by-bit. This allows the player to piece the storyline together themselves, while also participating in important events instead of being a passive observer.

It is difficult to achieve and requires a gifted team designing and building the game, but the player of a game with a gratifying story incorporated into the gameplay can undergo a certain type of catharsis that is an indicator of the potential of this medium for storytelling. This comes from controlling the protagonist's actions, and seemingly their choices, directly. The inclusion of challenging interactive elements is a very useful idiom for storytelling, not seen in other media. The player develops a vested interest in the main character in a way not seen in traditional media. Movies and novels can create an artificial emotional attachment to the main character to do this, and the emotional connection in computer games can be both more and less artificial. Less, because of the fact that a good actor can always create emotional resonance better than a computer-generated animation, but more because the player feels that they actually make a difference to the unfolding of the story.

All current examples of in-game storylines are pre-scripted, and the game world design and story construction is carefully done in tandem, allowing for sets of story events to occur independently of each other, yet only allowing the next set of events to occur after that group is complete. The new game, 'Half-Life 2' uses extremely sophisticated physics and emotional simulation to give the player a feeling of immersion in the game world, yet the story was meticulously planned side-by-side with the environment to enable a tightly controlled and compelling storyline. The writer of the Half-Life series, Marc Laidlaw, was deeply involved in the design of the gameplay, as well as the story, and this is what set the storytelling in these games apart. He mentions Edgar Allen Poe's 'totality of effect' as a guiding principle in his work [Laidlaw web]. This concept is related to the idea of the story as an organic whole, which Aristotle employed. The action elements in computer games are not, in fact, a completely separate element, but are linked to the storytelling in a mutually beneficial way.

3.2.2. Adventure Games

Three members of Lucasarts were instrumental to early successes in the graphic adventure game genre; Tim Schafer, Ron Gilbert, and Dave Grossman. This team's games were based on the previous text-based adventure games, yet the graphical and verb-based interface added a new level of playability to the genre. The skill with which these adventure games tackled the problems of in-game storytelling is very compelling. Every line of dialogue was pre-written and in the later games, was studio recorded, so the story had to be carefully constructed so as to make sense given a high level of player interactivity with the game world. Humour was used expertly to mask the limitations of the system, and the many useful mechanisms that were developed included the flexibility of the game engine, SCUMM, the use of puzzles that are important to the unfolding of the story, and strongly differentiated characters with a wide range of responses available, many of which were multi-purpose.



'The Secret of Monkey Island' by Lucasarts

Story Summary:
Play as an aspiring pirate and travel the Caribbean in search for the Secret of Monkey Island.

'Day of the Tentacle' by Lucasarts

Story Summary:
Travel through time to help Dr. Fred stop Purple Tentacle from taking over the world.

'Grim Fandango' by Lucasarts

Story Summary:
Uncover a conspiracy in the land of the dead as Manny Calavera, grim reaper extrordinaire.

Figure 3.2: Graphic adventure games

I have found it rare to find another game that tells a story as well as these old Lucasarts adventure games. They exemplify, to me, the skills necessary to tell a story as a dynamic interactive system. With games becoming more complex and taking longer to develop, the few game designers who can achieve this are taking longer to get their products to the public. Increasing the level of interactivity of game worlds also means that it becomes more of a problem to author consistent and entertaining stories.

Schafer is currently developing a game, 'Psychonauts', that promises to deliver a high level of interactivity and freedom within the storyline. He is a pioneer in the use of programming concepts for story dynamics, using procedural models of conversations, and numerous rule-based approaches that maintain a high level of humour and artistic expression [Schafer 2004]. The combination of the skills of programming and storytelling in this way is quite rare, which is why I believe future educational curricula should encourage it.

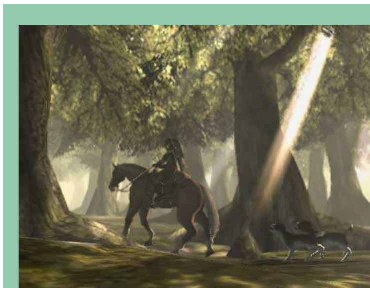
3.2.3. Hybrid Games

A large number of extremely successful action-adventure games exist, and game designers have developed a standard set of techniques that are becoming more prevalent. These games incorporate elements from both the adventure genre, such as characterisation and puzzles, and the action genre, such as reflex-testing combat. The player character is usually able to attack other characters in a varied array of ways, but there are also characters that can supply information or progress the story. There is also usually an inventory of objects or abilities that is acquired, and can be used to effect the environment, or be used on other characters. This standard *game mechanics* (as defined in chapter 1) setup is one that allows the world to be populated with entities and non-player characters (NPCs) that react in a scripted way to all the possible interactions a player has at his disposal, thereby leading on to new areas or abilities. In this sort of environment, stories with a large degree of interactivity can be authored.

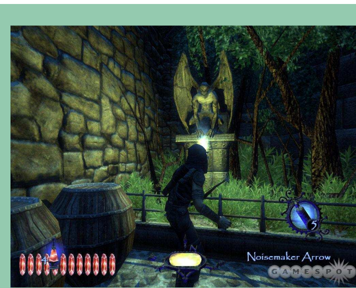
An outstanding example of all this is seen in the ‘Zelda’ series by Nintendo and Shigeru Miyamoto. Although each game in the series is distinctive, many game mechanics are common. One is the use of an expansive free-roaming 'overworld', containing a progression of ever more difficult dungeons. There is a clear difference between hostile and friendly areas and characters. This portrays the clear moral character of the player's hero character, because players will seldom question why some NPCs should be attacked and some not. The Zelda games also feature a very intuitive interface, which makes for more immersive play. The combination of technical achievement and playability seen in the Zelda series is an enviable achievement for a game designer, but the character AI involved is intentionally kept simplistic to further the story. To enable true istory in this type of game, however, the NPC AI would have to be more sophisticated, with a model of personality that could enable characters to take on either ‘bad guy’ or ‘good guy’ roles, depending on the player interactions. This would fundamentally effect the morally unambiguous nature of storytelling in a game like Zelda, and points to one of the biggest challenges istory developers face.

In the 'Deus Ex' series by Ion Storm , and the 'Thief' series by Looking Glass Studios , an approach called *emergent storytelling* was developed based on the concept of a storyworld as a complex system. Emergent storytelling can be defined as the creation of a storyworld with enough complexity such that an interactor observes a story emerging from their own interactions with the system, instead of through tightly defined and scripted events. The Thief games were based in an alternate mediaeval world where the hero, Garret, is a skilled thief who meddles in the local affairs of politics and magic.

A theoretical framework based on the interaction of the mechanics, dynamics, and aesthetics of the system was used by Looking Glass in the Thief series [LeBlanc 2004]. The result of this was that the player was given a set of goals, and was allowed to achieve them in myriad individual ways. Although the long-term story arc was authored as a series of atmospheric cut-scenes between stages, each stage allowed a relatively large amount of freedom in the achievement of goals. The freedom the player had in a stage was based on the design of the environment maps. Some stages featured a single path that led inexorably to the goal, but most were broad and allowed multiple paths, and different tactical approaches could be taken no matter which path was chosen. The player would interpret their own specific path as an element of the story, and could thus characterise the protagonist, Garret, in their own way.



Link in The Legend of Zelda by Nintendo



Garret in Thief: Deadly Shadows by Ion Storm

Fable by Lionhead

Figure 3.3: Action adventure games with a degree of emergent storytelling

Warren Spector of Ion Storm is a game developer that has pioneered the practice of storytelling as defining an interaction scheme with a system [Badman web], and worked on the Thief series with Looking Glass studios. Due to the constraints of the game industry, his game concepts do not usually make it fully to the finished product, but an aim in developing Deus Ex was to create an environment that was so fully realised and simulated that a story could emerge based on the player's interactions. This is going to become possible as storyworlds become more complex, and a greater level of understanding of emergent storytelling develops.

A standard axiom in storytelling, movies, and also in games, is that everything that happens should have relevance to the experience as a whole. In games, this usually entails a strict delineation between background elements and elements that are interactive, that the player becomes aware of as they play. The new game ‘Fable’ by Peter Molyneux's Lionhead studios attempts to allow for as much player interaction with the world as possible, and incorporates the player’s propensities in a feedback which makes the world react to every possible act by the player character. The characters in the game also remember player actions, and a gossip system is in place, which allows the player’s fame as a hero to spread around the world in a dynamic way. This, coupled with an authored story, gives the player an enhanced level of immersion in the game world. The choice to design an istory system for this sort of game would entail a story management system that expands on the feedback mechanism and decides on which characters to push forward as central characters in the story. Molyneux also touts the game’s flexibility in terms of the player’s ability to play the game as whatever type of character they want, in whatever way they want. This is extremely important for istory, and a system that allows the details of how the plot unfolds to be relevant to the player’s assumed character, is highly desirable.

It should be noted that the practise called *non-linear storytelling* is not discussed in this thesis, as it fundamentally remains as tightly controlled and scripted as linear models, but allowing for controlled choices on the part of the player. Also the term non-linear has different connotations in movie scriptwriter's jargon, and thus carries a weight of confusion when discussing an intersecting field such as istory.

3.2.4. Massively Multiplayer Games

With the enormous growth of the internet, the Massively Multiplayer Online Game (MMOG) has started to take a hold on players. Before this, games such as MUDs (Multi User Dungeons), between a small group of players with networked PCs, were a relatively popular diversion, but these are not the direct predecessors to the MMOGs. Most early MMOGs (Everquest , Ultima Online) were centred on the rule sets of tabletop RPGs, made famous with the Dungeons & Dragons series. The central activities of building and developing characters, exploration, and combat, have been transferred to the new domain. There is one early exception, 'Habitat' by Lucasarts from 1985 (then called Lucasfilm games). 'Habitat' was the first MMOG ever, with thousands of concurrent players within the same gameworld, and a 'post-mortem' analysis of the development of the society that sprang up in its world [Morningstar and Farmer 1990] is a fascinating introduction to the problems encountered by the developers of such a project. As Habitat was initially a very free society, some players decided to wreak havoc by causing crime and disrupting other players' enjoyment. This is a common problem in MMOGs, usually resulting in the offending players' ejection from the game. However, the developers at Lucasarts decided to let some matters be settled in-game, by having a virtual 'trial' of the offender [ibid]. The resulting inclusion of the punishment mechanism as a part of the game experience led to a greater feeling of society and many players were very happy with the results.

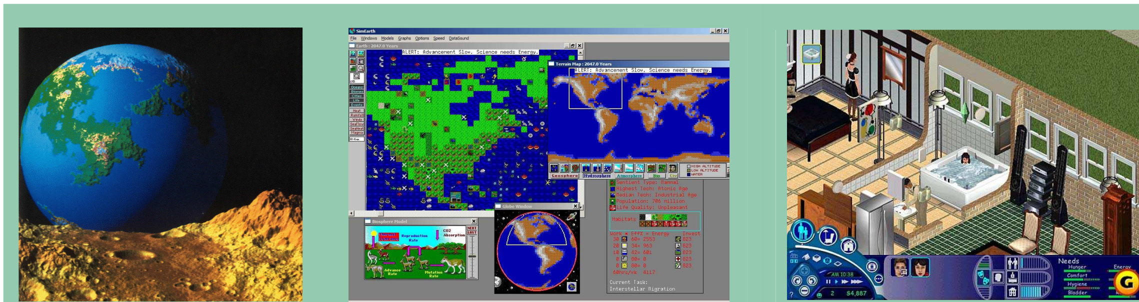
This example is a compelling case for allowing players to take any attitude they want in the online game world, so long as actions are responded to appropriately. In order to maintain Habitat, a team of developers had to continuously monitor events and respond to players' actions, taking the equivalent role of the 'gamesmaster' player in tabletop RPGs [ibid]. This is still the normal model, with the notable exception of the 'Active Worlds' game, which was bought, and was maintained, by the player community after the company expired.

MMOG gamesmasters have a number of standard mechanisms to tell stories. The designers first create an extensive back-story, with a view to allowing the thousands of players to each discover story threads according to their own choices in the game. The currently active gameworlds in 'City of Heroes' and 'Star Wars Galaxies' take this model, with myriad possible areas and NPCs for players to explore. A common mechanism is for players to talk to NPCs to receive quests, with the quests not having any influence on the overall story arc. This is a problem that could be rectified with a more in-depth simulation of the NPC society, whereby the outcome of player activities could have long lasting effects on the overall story arc that the gamesmasters author. Massively multiplayer games is an extremely complex area of development and study, and although I will later put forward a case for the possible usefulness of a system like OPIATE in such an environment, the complex theory and practise of MMOGs is for the most part outside the scope of this thesis.

3.2.5. Simulation Games

The creation of content for computer games has become a bottleneck in the development time of new titles, and a new approach is being touted as the way forward; automatic generation of content [Carmack 2004]. This will not consist of purely generational, but derivational algorithms, which work with a library of previously designed elements, and create new variations on them. The use of middleware is becoming a more popular way of increasing a game's appeal, with Havok's physics engine in widespread use in current and next-generation games (see www.havok.com), and systems such the SpeedTree's flora-generating engine gaining acceptance. The conception that every object need not be placed in a level by a designer, but is part of the game system, evolved using a software model, enables the creation of larger, more complex worlds. Work in fractally generated landscapes [Mandelbrot 1982] shows how extremely complex geography can be stored in relatively small files, and generated when required. Mandelbrot's work has been dramatically extended by Musgrave [Musgrave and Mandelbrot 1992] to create whole simulated planets that have fractally generated clouds, landscapes, water, and more (Figure 3.4). I think that automatically generated content coupled with simulated game worlds will enable players to delve into experiences of heretofore impossible interactivity.

The simulation of various systems in games has been pioneered by Will Wright and Maxis studios. Wright is a game developer who regularly employs simulation methods to great success in his games, such as SimCity, SimAnt, SimEarth, and The Sims. SimCity and SimEarth used layered cellular automata implementations, where each layer had feedback loops between the layers above and below it, based on simple rules of interaction [Wright 2003]. Early simulation methods, such as Conway's *Life* implementation, inspired this paradigm. Wright based the rules in SimEarth on the theory of Gaia, developed by James Lovelock [Lovelock 1987], with interlocking geosphere, hydrosphere, atmosphere, and biosphere models. The rules of SimCity are based on simple economic and social models, yet each time the game is played, the outcome is completely different. An example was shown in Wright's GDC 2003 lecture [Wright 2003], with the relationship between crime, land value, and population. A common objective in simulated game systems is balance, with desired equilibrium achieved when opposing forces are mediated by manipulating variables of the system.



Fractally generated Landscapes

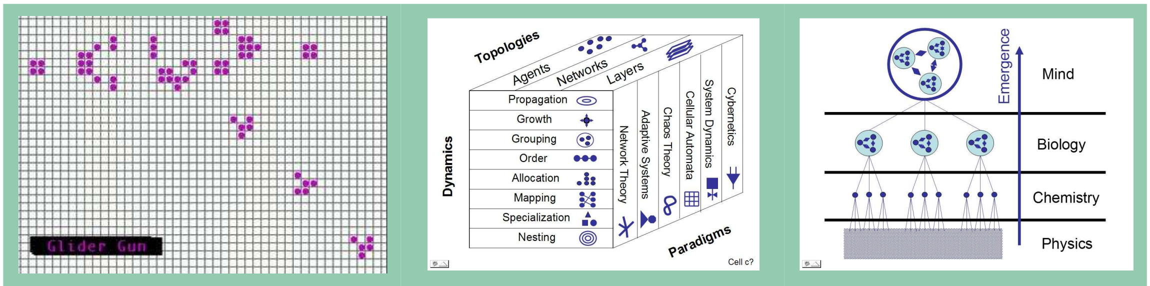
SimEarth by Maxis

The Sims by Maxis/EA

Figure 3.4: Generation & Simulation

With The Sims, the most successful PC game ever, Wright used his skill in simulation to model human interaction using a remarkably simple model of desire and gratification. All the objects in the gameworld contain information about what desires and needs they gratify, and each character has a dynamic set of needs and desires. Wright is clear on the fact that his model does not facilitate interactive storytelling, but that the players can interpret the system dynamics as stories. He recognises the intrinsically linear nature of stories, compared with the open-ended experience in playing with simulated systems.

Wright's definition of emergence is one of the clearest and intuitively satisfying I have come across, show in Figure 3.5 below. Emergence occurs when a complex system gives rise to a perceived pattern at a higher level than the rules of the system operate at. The higher-level system operates with its own rules and patterns that are implicit in the definition of the lower-level system, yet are not explicitly modelled in the system, and can thus sometimes give rise to unpredictable behaviour.



A glider gun in Conway's *Life* | Topologies, Dynamics, and Paradigms | Emergence

Figure 3.5: Taken from Will Wright's GDC 2003 presentation

It has been postulated that defining the rules of a simulated system can be a medium for artistic expression [Musgrave 1995]. The implementation of emergent simulated stories will entail the simulation of a system that does not yet have corresponding scientific laws, as physics or even social science do, as these story systems seem to exist only as static representations – a novel for example, rather than processes. However, the literary structuralists have provided a starting point for the development of such a set of rules, and some game designers have implemented systems, not with an eye towards the traditional modelling of reality, but modelling a system that provides interaction that is interpreted as dramatic. In the Thief series, Looking Glass studios developed a set of rules of behaviour that resulted in dramatic encounters, with guards and other characters. This was an example of the design of a set of mechanics that resulted in emergent dynamics, which itself resulted in dramatic aesthetics. For the sake of example, I have reproduced below a story told by a player describing his interpretation of one of the missions in Thief (Figure 3.6). This is one of many that fans have posted on the www.ttlg.com (through the looking glass) internet message board.

I was making my way up that weird mage tower, and I kept feeling like I was getting further and further into places I shouldn't be, and I finally found the last secret and got to the very top. There was the evil glowing book with the candles around it, and naturally I went right up to it and opened it. I saw the wierd diagrams and strange language, and when I closed the book, I heard "...meeerghh...." from the darkness behind me, and my heart skipped a beat (because I knew exactly what it was and it was the last thing I wanted to happen) and everything from that split second on was in slow motion...

Somehow my subconscious knew where the nearest "fastest exit" was, from the quick look around when I entered, and it wasn't the elevator, it was the window across the other side of the room. I remember turning around (this seems like in slow motion, remember) and seeing the window come into view and some silhouettes of not one but TWO zombies. I immediately moved forward at full speed (in slow motion) running right past the nearest zombie, and jumping in perfect timing to land on the window sill like a cat.

Then time returned to normal speed when I got a hold of myself for a brief moment (a BRIEF moment, mind you) when I saw the humungous drop down to the streets below, which made me turn around and second guess my action. ONLY for a second though - I instantly turned back around and RAN straight off the edge, experiencing the long fall down, and hitting escape EXACTLY when he made contact with the ground.

That is the first, and ONLY time, a game has ever scared me so much that I actually INTENTIONALLY committed suicide in the game rather than experience what might ensue!! It all happened so fast, I barely remember making the decision.

The recounting of the 'Life of the Party' mission in Thief 2 by
Domarius (Clint Hobson) at www.ttlg.com

Figure 3.6: An Aesthetic interpretation of the mechanics and dynamics of Thief 2

A set of mechanics which result in an aesthetically pleasing emergent storyline has not yet been designed, but I believe the use of plot-level simulation in a simulated game world could result in the emergence of unauthored stories. One thing that I would note in this is that through the process of choosing the rules or mechanics of such a system, the type and nature of the resulting story is influenced. This shows that there will not be any single 'law' of story simulation that is accepted as true, as there are as many types of storytelling as there are people to tell them. Thus, we may see a set of rules and dynamics that result in a type of action-cop-adventure story, and a different set that result in the emergence of dramatically satisfying romance stories.

Some successful games storytellers have used the concept of keeping the player character the focus of the story [Schafer 2004], [Laidlaw web]. Thus, in creating stories for players, every character in a large gameworld need not be constantly simulated, as the focus of the story is on the player character(s). The NPC characters that are not engaged in the current situation of the hero need not be involved, so the simulation rules should be dynamically applied to the relevant NPCs. In the Ultima series of RPGs, Origin studios developed a mechanism for making animals and NPC characters seem more real, by flagging certain entities as natural enemies, so when the player encountered them, they started attacking each other, making the game world seem like it had an existence independent of the player. This approach could be extended for an istory system so that the plot-level simulation works on a higher layer than the basic world simulation, and its focus follows the movement of the player character around the world, and tracks the currently important NPCs. The use of paradigms from AI research is instrumental in the development of such an adaptive, focused, simulated story system.

3.3. Artificial Intelligence

GOFAI, or ‘good old-fashioned AI’, has been developed in the past few decades as a means of solving certain defined problems. The travelling salesman problem is one classic example, where an optimal route between a set of cities is calculated such that the route does not cover previous paths taken. Scheduling problems, information filtering, planning, classification, and pattern-matching are other areas that have been addressed. Different techniques have developed for each type of problem. Artificial neural networks (ANNs) have been found to be good at pattern recognition, and genetic algorithms (GAs) are good at finding near-optimal search solutions efficiently, but progress has been slow in developing a system that tailors its approach to dynamic, multi-domain problems automatically.

One of the difficulties involved in building an AI which works in an area that is seen to require creativity, lies in the nature of creative thinking. The ability to transform and apply diverse types of solutions from areas that may not immediately be obvious to new problems is a feature of human creative thinking [Wiggins 2001]. Some progress in analogical reasoning has been made in Maynooth College [Crean 2002], which could enable an AI to match certain structural features between problem domains, allowing it to apply solutions that are relevant to a previously unseen problem domain. There are a number of research programs with the goal of producing artificial general intelligence (AGI), notably the Novamente project [Goertzel et al 2003], the CYC project [Stork et al 1998], and others. It has been recognised that to create an AGI requires the use of a complex architecture that incorporates a hybrid of the GOFAI techniques, allowing it to apply appropriate problem-solving techniques to different problem domains. This section will be a description of how different strands of AI research are applicable to the general set of problems relevant in story generation and presentation using an artificial simulation of intelligence.

3.3.1. Rule-Based AI and Expert Systems

An early method of implementing systems that appear intelligent was simply the use of a set of logical rules that encode the knowledge that a human expert would use in a particular domain. Although limited, and little substantial new progress has been made recently, their efficiency means that rule-based systems are still in use in a wide range of domains such as help desks [Barr and Rewari 1995], medical diagnosis [Berner 1998], and crop management [Rafea 1991]. Some domains, where the expert knowledge applies to the whole problem space, are appropriate for a rule-based approach. New knowledge can be extracted from the system in one of two ways, forward-chaining and backward-chaining. In forward-chaining, a set of assertions is given to the system, and the rules are applied until the system concludes with any logical results, while in backward-chaining, a set of assertions are given to the system, and the requirements, that must be present for those assertions to be true, are output as the result.

Although rule-based systems cannot learn, it has been shown [Craven 1996] that more advanced AI methods that *can* learn can be trained and then the state of that AI can be re-composed as a rule set to optimise efficiency. The rules that are thus generated solve the same problems that the AI has learnt to deal with based on experience, yet perform much better in real-time applications. Rule based systems are most useful when combined with other AI paradigms, and this can be done using many possible architectures, for example in [Holland 1986].

3.3.2. Using GAs, ANNs, and CBR

The biologically inspired AI paradigms of genetic algorithms (GAs), artificial neural networks (ANNs), and case-based reasoning (CBR) are disparate approaches to problem-solving, each allowing learning and adaptation. They respectively correspond to the human cognitive activities of trial-and-error, pattern-matching, and memory-based reasoning. They all conceptualise a problem space as an N-dimensional set of variable features, with problem descriptors occupying a single point in that space. The problem is input to the system, and a solution to the problem is the output of the system. An evaluation of the new solution can be fed back into the system to facilitate learning. This mechanism requires feedback on the success of the solution so that solutions can be evaluated. The most common datasets require classification solutions, and the AI can be trained in a supervised way based on user feedback, or in an unsupervised way, so that it decides which solutions are correct based on a heuristic which is calculated based on some user-defined equations.

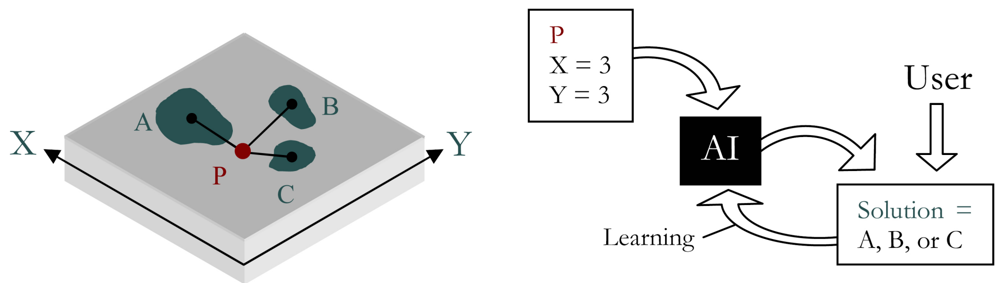


Figure 3.7: A General AI Problem space & a Learning mechanism

In Figure 3.7, we see a problem descriptor, P, in 2-dimensional problem space, which contains 3 clusters that represent classifications of previously encountered problems, A, B, and C. P is closest to the centre of cluster C, but closest to the edge of cluster A. This is the sort of problem that AI researchers can take diverse, domain-specific approaches to, and there is a parallel between the way AI systems are developed to account for them, and the way commercial computer games are developed. The use of experimentation and an organic design process, where design and implementation are done in alternating, or parallel, sequences [Carter 2001], is common to both. There are strategies, yet for the most part, there are not defined algorithms, that are common across a large range of problem domains. Therefore research in artificial general intelligence (AGI) is attempting to find ways of abstracting general methodologies that can transfer across domains [Goertzel 2003].

A problem with using AI for solving problems requiring creativity is that of encoding a heuristic which can evaluate solutions for the domain. Humans solve problems creatively using complex internal models of the problem domain, working on a number of levels simultaneously to achieve a solution that balances many requirements. The requirements inherent in storytelling mean that all individual solutions (plots) are usable in some particular context (theme and setting) of the problem (producing a plot and enacting it). In this domain, case based reasoning is a good choice for plot composition, as all the cases in the case library remain available, and instead of generalising over the whole problem space, solutions can be recalled that fit the context, enabling the operation of the AI in a dynamic problem space. Section 4.6 of this thesis will show how the OPIATE system's case-based component performs these tasks.

GAs and ANNs can 'overfit' to areas of their problem space that they are most familiar with, meaning that when new problem contexts are presented, the AI may get stuck trying to use inappropriate solutions, although there are mechanisms to avoid this. However, CBR is a better choice for the domain, as I believe it represents solutions in a way that is conducive to the representation of story plots. This is because CBR allows local adaptation in a way reminiscent to that done by oral storytellers, where a story can be minimally altered by the narrator to allow for audience preferences, while preserving the structural and tonal quality of the story.

3.3.3. Case Based Reasoning

Case-based reasoning is a newer paradigm than GAs and ANNs, yet has fast become as well-known and well-regarded. Initially introduced by Schank [Schank 1982], there is now a wide range of problem domains CBR is used in, most commonly help-desks and decision support [Bergmann et al 2003]. A case library of previous problems is stored and used to solve new problems, mimicking human psychological models [Tulving 1977] of memory-based problem-solving. A demonstrated advantage of CBR over other methods is the efficiency of solving problems identical to, or similar to, those that have previously been encountered [Watson 1994]. The design of a CBR system consists of five distinct areas; **Representation, Retrieval, Reuse, Revision,** and **Retention**. The CBR cycle is shown in Figure 3.8.

- **Representation** - How the system represents its knowledge. CBR systems often store cases as problem/solution pairs. These can be indexed to facilitate efficient retrieval at run-time.
- **Retrieval** - When a new problem is presented to the system, the case with the closest matching problem descriptor is retrieved. Multiple cases can be retrieved, and combined for use in the solution.

- **Reuse** - The retrieved solution, or set of solutions, is adapted to the current problem, using domain-specific or context-specific knowledge.
- **Revision** - The resulting solution is revised based on feedback from the user, or from the success heuristic. This revision can be an adaptation procedure.
- **Retention** - The new problem/solution pair is stored in the case library for future use.

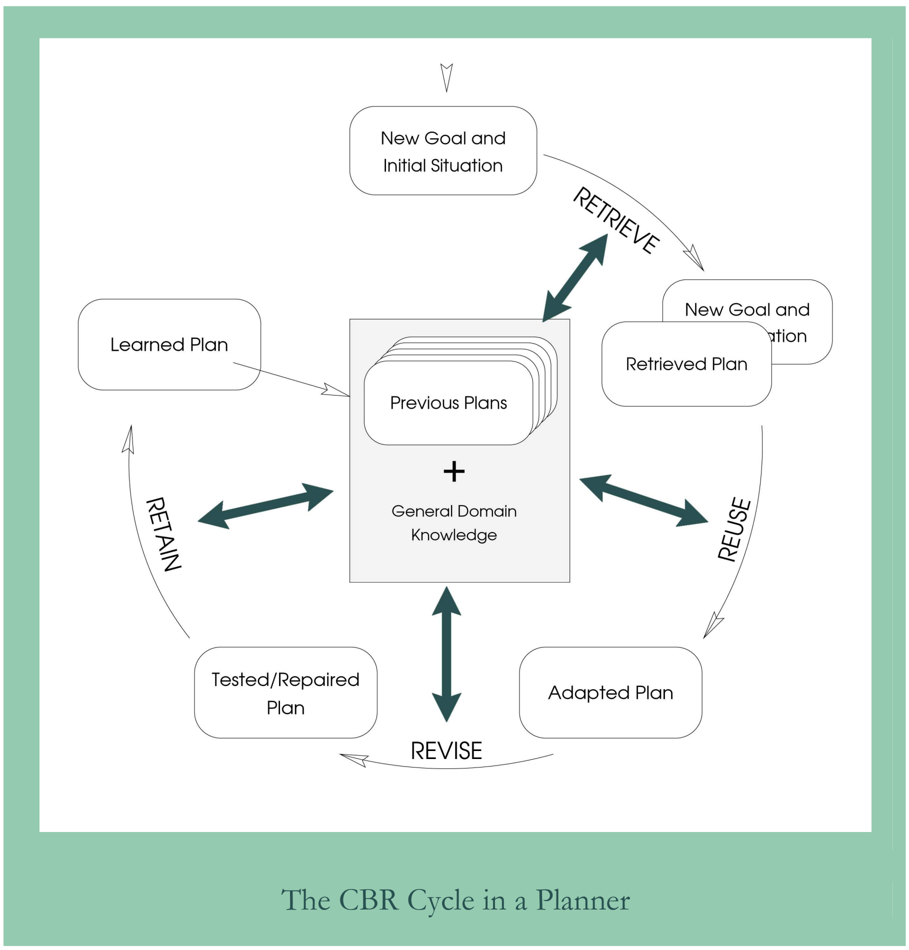


Figure 3.8: The CBR Cycle

A number of researchers have contributed to the acceptance of CBR [Veloso 97, Muñoz-Avila 2003], overcoming various problems that have been encountered. One mechanism for enabling greater generalisation capability is the use of skeletal cases [Bergmann 1991], whereby the case library consists of abstracted knowledge, and a retrieved case is adapted to new problems by manipulating features of the case to fit the context of the problem. The newly-generated case is then converted into a skeletal case for storage in the case library.

For the MINSTREL storytelling system, Turner [Turner 1992] used a complex case-based reasoner which integrated what he terms 'episodic memory' and 'imaginative memory', which allows a recursive case-based architecture, where modules of the case-based planner can themselves use case-based 'imaginative' reasoning to adapt cases. His system is an early example of advanced AI techniques in a domain requiring creativity, inspired by Meehan and Lebowitz's previous attempts [Meehan 1976, Lebowitz 1983]. His work proved extremely successful in telling stories within the genre and setting of Arthurian legends, creating novel plots and motifs based on a user's input of theme and setting. MINSTREL was also capable of applying cases from the Arthurian domain to the domain of the modern soap opera, as it could adapt the concepts of knights, horses, and princesses, to men, dogs, and girls next-door. The MINSTREL system will be more fully described later, in section 3.3.9.

3.3.4. Planning

Formally, planning has been defined as the construction of a set of operations, organised temporally, that will transform a given initial situation to a given goal situation [Geffner 2002]. The field known as classical planning attempts to search the space of possible operations given the initial situation until a plan that results in the goal state is found. This can be a very complex operation in large problem spaces, therefore is not applicable to a real-time complex planning system.

Planners that integrate case-based reasoning have been found to significantly reduce the complexity of the planning process [Mali 1999a]. Plans are formed through the application of appropriate recalled plans, and the adaptation process is a local search that begins from the retrieved plan. The system re-constructs the plan based on an analysis of the features of the goal situation and initial situation. Pollack et al [Pollack et al 1994], use plan filtering to show how focused planning increases the timeliness of real-time planners. Veloso [Veloso 1997], along with Mali [Mali 1999b], show how plan merging can help case based planners perform analogical mapping to deal with new problems, and Muñoz-Avila et al [Muñoz et al 1999] employ hierarchical planning to control a multi-agent system in the IMPACT environment [Dix et al 2002].

Veloso, Muñoz and others have developed an extensive set of conceptual tools that perform extremely well, and Veloso and Rizzo have applied some of the techniques used in the PRODIGY planner to storytelling AI [Rizzo et al 1999]. A character-based approach is used, and the goal is to create believable characters, by giving them intentions, goals, and plans. R. Michael Young uses planning for story generation in the Liquid Narrative group [Young 1999], where an environment-independent story telling AI architecture has been in development, with a successful implementation in the Unreal Tournament engine completed. Research in believable agents was initiated in the Oz project in CMU, spearheaded by Joseph Bates and his group [Bates et al 1992], which has since become a business called Zoesis, who create believable agent-based computer games.

3.3.5. Agents and Situated AI

The term 'agent' is one that has many uses in computer science. There has been put forward a number of different sets of criteria that a piece of software must meet to be regarded as an agent (see [MacNamee 2004] for an overview), and a number of different types of agent have been defined, such as intelligent agents, autonomous agents, deliberative agents, and believable agents. The consensus seems to be that an agent is an encapsulated component that has an internal model of their environment, a set of beliefs about the environment, and desires to alter that environment, along with the capability to do so, epitomised in the standard BDI (Belief, Desire, Intention) model.

There is a clear difference between an AI implemented using one or more agents that inhabit an environment in real-time, and an AI that exists as a central algorithm with no experiential knowledge of its environment [Goertzel web]. Sims[Sims 1994] and Azuaje [Azuaje 2002] have developed compelling demonstrations of agents that have learnt surprising ways to achieve goals, using an evolutionary approach. Although the individual agents do not learn, the genetic evolution of their topology and 'minds' enable unpredictable behaviour and topology to evolve. Because in Sims' implementation the physical topology of the agents dictates the arrangement of their cognitive structure, there is a more life-like result in the agents' movement and behaviour. The behaviours are learnt by pitting one creature against another in the race to reach a goal, and over many generations, the most effective topologies and corresponding cognitive behaviours are mutated and cross-bred. The environment in this case is very simple, but Sims's approach holds much promise for more complex simulations (Figure 3.9).

Figure 3.9: Evolved Agents

Agents in dynamic environments need to be able to adapt. Whether the learning is done with genetic adaptation, or machine learning algorithms, an agent must be able to react to new situations intelligently. The concept of situational intelligence [MacNamee 2004] is a new paradigm in the development of agents. MacNamee used the paradigm of virtual humans and emotional simulation to show how a complex society of believable social agents can help immersion in computer game environments using a model of situational intelligence that adapts as needed to the dynamic situation. However, the term 'agent' usually refers to embodied individual agents, and while this research is important for creating istory, my research follows the lesser-trod path of modelling a real-time AI story director as an agent. An omniscient agent that inhabits a simulated environment, but whose influence is limited to effecting the semi-autonomous agents that populate it, is the general approach that has been taken in my work. This general approach has been recognised as promising, but it has been attempted in relatively few projects, although Weyhrauch in CMU made an initial investigation into real-time plotting [Weyhrauch 1997], Szilas [Szilas 2002] and Sgouros [Sgouros 1999] have made important contributions, and currently Magerko [Magerko 2002] and Braun [Braun 2002] are working on systems similar to OPIATE.

3.3.6. Artificial Immune Systems

The functioning of an Artificial Immune System (AIS) is based on a biologically-inspired AI paradigm that has elements and mechanisms in common with neural networks, genetic algorithms, case based reasoning, and adaptive systems, among others. It is a new and growing field of study that uses models of the biological immune system to create problem-solving AI systems. Applications include robotic control [Krohling et al 2002], hardware fault tolerance [Canham and Tyrell 2002], and virus checking [Forrest and Perelson 1994]. The benefits of using an AIS are those of the above-mentioned classical AI paradigms combined, yet in its combination of the various paradigms, it appears to have lost some of the weaknesses inherent in them. Although an AIS was not implemented for the OPIATE system, the concepts relevant to story generation in a more general sense are suitable for discussion in the context of AISs.

Propp’s usage of biological terms such as species and genera in his morphology, and his discussion of stories as entities that undergo evolution and adaptation much like organisms, invites the use of biologically inspired AI in the generation of stories. Indeed, Richard Dawkins' concept of a meme [Dawkins 1976], and memetic evolution, conceptually fits extremely well to Propp's model of the folktale plot. Thus, a modification of the genetic adaptation algorithm is envisioned whereby the physical agents occupy one 'layer' of a simulation. Another symbiotic layer of simulation concerning the dissemination and mutation of stories of the agents' experiences can then occupy another layer.

The requirement for the agents' survival is based on how well they fare in the world, and the requirement for the stories' survival is based on an independent evaluation of success. Even if an agent fails to 'breed', stories of their successful experiences can proliferate, if the agents disseminate stories of their own experience, both first and second hand, and use the stories that they know about to help determine how they act in the world. This simulation could act as a crude model of human culture. This architecture could also be used for an interactive game world, and would be a major improvement in the believability of virtual worlds.

AISs have close correlations with the functioning of the system outlined above. They incorporate memory of previous encounters, and new agents are continuously generated to deal with the dynamically changing problem space of pathogens. In the biological immune system, pathogens are bacteria and viruses that cause harm to the body, yet there are also bacteria that are complementary to the functioning of the body. A key mechanism is the detection of non-self elements. These elements are assimilated, and the assimilation process facilitates the system’s learning. Pathogens are in fact mostly benevolent towards the immune system as a whole, as they enable the evolution of a hardier immune system, this being the motivation behind infant inoculation. B-Cells and T-Cells (white blood cells) are continuously generated by the bone marrow, especially at times of infection. Although immunologists are still not in full agreement about the functioning of T-Cells [Kim and Bentley 2002], B-Cells are known to be the chief mechanism of distinguishing non-self elements, and fighting harmful entities [Neal 2002].

Even if it is not biologically plausible, although there is some correlation in biological 'helper T-Cells', I postulate here that the T-Cells could be seen as mechanisms for communication between the B-Cells. As old B-Cells die, and new B-Cells are formed, the most effective B-Cells are lost, and their abilities to neutralise pathogens are also lost. If the T-Cells act as 'teachers', sensing and storing the most effective B-Cell configurations, and passing on this information to new B-Cells, this can serve as a type of long-term memory for the system. If this was implemented in an AIS, one observed problem, mentioned in [Neal 2002] could be countered. Here, it was observed that in the time-varying data space of Fischer's Iris data, long-term memory of the system was compromised by using conventional AIS mechanisms.

If the concept of a pathogen is stretched and moved to the analogous area of the dynamics of story simulation systems, so that it represents a new 'villain' or systemic threat, the collective ‘mind’ of stories reacts to that pathogen through one of two basic responses. Firstly, old stories are used on the pathogen and if the action suggested by that story is successful, the stimulation level of that story is increased, and it is more likely to be passed on to new agents. In this model, the stories themselves act as the 'teacher' cells mentioned above. Stories could be mutated or adapted every time they are passed on, to fit with the particular agent's setting.

Secondly, a new 'naïve' agent (from the biological terminology) can create its own story through its actions and the reactions of its environment. New actions by naïve agents could be generated through a randomly seeded 'personality' model for the agent, or from crossovers between agents. This mechanism generates new stories, yet because the agent can apply its second-hand experience of stories to creating the new story, it creates something not entirely different and new, but based on the structural and contextual templates it has come into contact with. These mechanisms could conceivably be implemented in an online multiplayer game that evolves a story library based on the interactions of its users. The stories would be proliferated by the NPCs and the stories could be about other players and other NPCs. Here, Levi-Strauss's 'Myth Mind' is explicitly modelled as a simulation of memetic stories, as the system of T-Cells in an AIS-like system.

Although these mechanisms were not implemented in the OPIATE system, the introduction of AIS systems here is more appropriate to a general AI section, and I have chosen to put it here. The OPIATE system utilises a case-based system coupled with a population of semi-autonomous character agents, which conceptually looks like an AIS where the story memory facilities are implemented centrally, instead of as discrete T-cell like agents. The ideas described above will also relate to some future work suggestions in chapter 7, specifically in section 7.6.

3.3.7. Creative AI

The creation of novel and original solutions to problems by an AI is difficult to facilitate, and to this end a number of researchers have developed ways to allow AIs to recognise their own efforts, and evaluate them [Cardoso and Wiggins (ed) 2002]. Usually, implementations of creative AI are very domain-specific, and the authors of these systems have in fact injected their own creative methods into the functioning of their systems. Human creativity seems to work through the combination of ideas more often than the creation of completely new ideas. This is how most AI algorithms work, either through stochastic permutations of ideas, or combinations based on mapping functions between domains and contexts. Concepts and features useful to the AI are encoded in the definition of the system, and the adaptation and evolution of the system is the process of combining elementary components until a satisfactory solution is found.

Authoring creative AI behaviour is easier than creating a system where creativity is seen to emerge based on the dynamics of a system. The emergence of creativity requires that the AI be situated in a complex dynamic environment, capable of discovering knowledge itself, based on a design that allows emergent system evolution [Bentley 1999]. Saunders and Gero have done some work in creative social agents, where creativity is seen to emerge in a social simulation, where cliques and other social structures emerge [Saunders and Gero 2001].

Mateas takes the point of view that AI algorithms can be used in artistic expression by the programmer-as-artist [Mateas 2000], leaving the onus on the programmer to be creative with the use of AI techniques. His research shows how an AI effectively increases the scope of the artist's vision, into the interactive behaviour of the artwork and it's capability to respond to audience participation. AI systems that respond to their own output have shown that novel and interesting creations can emerge, using simple enough models, as seen in the 30-year song machine by Bordeaux et al [Bordeaux at al web], among others. As AI and Art have certain goals in common, both being an attempt to express the human condition, the use of AI has been embraced by artists. Mateas [Mateas 2002] quotes Joseph Bates to eloquently express this: "It can be argued that while scientists may have more effectively created [machines that act like] scientists, it is the artists that have come closest to understanding and perhaps capturing the essence of humanity that … AI researchers ultimately seek" [Bates 1994]. However, it can be argued that rather than creating a simulation of humanity, many AI researchers, myself included, see AI systems as tools for human expression (artistic and otherwise), rather than individual entities that express themselves.

3.3.8. AI and Games

Game theory is an extensive research area in AI with application to such games as Go and Checkers, where all possibilities of the game state are searched and filtered. However, these techniques are not applicable to istory, as in complex storyworlds such as those expected by modern gamers, the task of searching all the possible story states and evaluating them is far too difficult. This is illustrated in Figure 3.10, taken from Brenda Laurel's 'Computers as Theatre' [Laurel 1991], which shows the 'flying wedge of possibilities'. The white wedge represents the dynamics of possibilities. As the story progresses, there are fewer and fewer possibilities based on the history of events, but more and more potential actions that could have been taken if the events had been different. The satisfying conclusion of events becomes more and more obvious to the audience, until only one possible outcome exists. If the player is free to effect the outcome of the plot, taking account of the player's actions in an ongoing simulation of the story could be a more efficient method of istory generation than a continuous search of all potentialities that could occur down the road. A mechanism for pre-empting all possible player actions is thus not necessary.

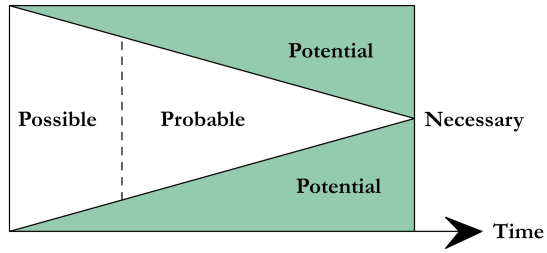


Figure 3.10: The Flying Wedge of Possibility

There have been a number of commercial games with interesting AI implementations from the academic point of view, although none use AI for tackling the wider problems associated with storytelling. In the game industry, the term 'AI' refers to the often simple and static mechanics used for controlling NPC behaviour, something quite different to its meaning in academic circles. Finite state machines are the most popular means of implementing NPC AI [Rabin 2002]. Even so, games such as 'Creatures' by Creature Labs, 'Galapagos' by Anark, and 'Black and White' by Lionhead, used learning algorithms to give the player more freedom, yet in fact, 'Black and White' did not use neural networks or any truly learning AI, but pre-defined behaviours that were 'switched on' when the creature in question was supposed to have 'learned' them [Rabin 2002]. Creatures, however, used an implementation of ANNs, divided into specialised 'lobes' within each creatures 'mind', and random genetic mutation for generating new creatures [Grand et al 1996]. The gameplay was very open-ended, more like a 'toy' than the goal-oriented gameplay in most games. Much like 'Black and White', the user explicitly taught the creatures through the use of reward and punishment interactions.

'Galapagos' was a little-known yet compelling game, which found a way of incorporating a learning AI with goal-directed gameplay. The player controlled a small robot that had to navigate through dangerous areas, hitting switches and overcoming traps. The player did not control the robot directly, but taught it, with reward and punishment methods, to use appropriate care in its exploration, eventually recognising safe and dangerous parts of the environment. Although Creatures and Galapagos are very interesting projects, they are rare examples in the game industry. There is a definite mistrust in the game industry of academic techniques, and there are good reasons for this, as it is a lot harder to exhaustively playtest these games, and no matter how advanced an AI is, it must be fun to interact with it. There are other examples of the use of AI in games, seen in 'mods' for games. The modding community takes existing games engines, and creates freely available modifications to them. Some notable mods include the SoarBot [Laird 1999] and NeuralBot. These respectively used a rule-based system and a genetically adaptive ANN system to create opponents for the first person shooting game Unreal Tournament.

Figure 3.11: Games with Learning AI

There are currently some projects in development that promise to increase the growth of AI in games. John Laird's paper [Laird 2000] on the use of games in creating human-level artificial intelligence has been well received in academia, with a number of game AI groups and courses having emerged.

There is still a prevalent antipathy towards allowing an AI to have control of the story in a commercial game, and as games have gotten more realistic with improved graphics, and the pre-scripted stories can still be more and more engrossing, many feel that leaving the storyline up to an AI would be an overestimation of its capabilities, since good storytelling is a difficult enough task for a human being. This is perhaps why the push in this direction is coming from academic projects, as they do not have to face the rigourous quality testing that commercial games must go through in the extremely competitive market.

3.3.9. AI and Stories

The AI necessary for a real-time istory-telling system is different to that used in most traditional AI domains. The success of a story is *very* subjective, and even if a story is widely accepted as a good one, finding an algorithm that evaluates this is very difficult, as stories can be good for a variety of seemingly conflicting reasons. For example, in a genre such as fantasy, or cartoons, the realism of the setting is not a good heuristic for success, yet in a drama where the realism of the characters' behaviour is vital to the story, it is.

People pick up cues in the setting and initial presentation of stories and intuitively know about what sort of events will transpire. Some modern storytellers take advantage of this, and genre cues can be deliberately misleading. An example is Quentin Tarantino's "Kill Bill", where the heroine, Kiddo, enters the house of Bill, the final villain, and finds him playing happily with a young girl. The story leading up to this is very violent, and every previous encounter concerned Kiddo's desperate struggle to stay alive and enact her final revenge. The 'end boss' encounter in this case deliberately jars the expectations of the audience, and this is not by any means a bad thing. The predictability of stories is a fine line on one side of which lies formulaic, unoriginal, tales, and on the other side of the line, a story can degenerate into a nonsensical, disparate, series of events. Thus, a satisfying level of predictability is one that keeps an audience on their toes, so that they never know which preconceptions will be shattered and which will hold true. Although storytellers in established modern media continuously find new and original ways to tell stories, building an istory computer system will necessarily consist of emulating older and more formulaic methods.

There have been a number of projects in AI story generation as static text. Alan Dundes' early implementation of Propp's morphology [Dundes 1965] worked by randomly stringing permutations of previously defined character functions together based on the rules Propp outlined. In 1976 James Meehan's TALESPIN storytelling program [Meehan 1976] used a goal-based character planning paradigm that still has influence in some of the current istory projects. Characters were defined, given beliefs and goals, and the 'story' that was produced entailed a description of how the characters interacted to achieve their goals. UNIVERSE built on Meehan's approach, but used a set of author-defined story goals to construct plot-fragments [Lebowitz 1983], which in turn each prompt characters to act them out based on the characters' goals. The emphasis in UNIVERSE, and later in Dehn's STARSHIP [Dehn 1989], was on the achievement of author-level goals based on an evaluation of the current story situation. Turner's MINSTREL [Turner 1992], introduced in section 3.3.3, was the most elaborate example of this approach. MINSTREL is a sophisticated case-based planner that constructs stories such as the one below in Figure 3.12.

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 so that he could fight a dragon. Grunfeld moved 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.

MORAL: Deception is a weapon difficult to aim.

Figure 3.12: Sample Output from MINSTREL

Turner used a number of innovative techniques to facilitate storytelling. His core method, inspired by the case-based reasoning model, was to use flexible Transform-Recall-Adapt Methods (TRAMs). Turner defined many of these, from the 'Generalise-Role' TRAM, for adapting previous plot fragments to new characters, to the 'Use-Magic' TRAM, for solving character problems using magic objects, and the 'Cross-Domain-Reminding' TRAM, which enabled plots from one context to be used in another. The TRAMS drew on a model of creativity that used 'imaginative memory' to gradually adapt solutions using incremental adaptations. Turner notes that he was initially inspired by Propp, but none of Propp's morphology was to be found in MINSTREL by the time it was finished. The program's models of theme, and story morals, and its ways to express themes using author-level goals, were basic to the system, and incompatible with Propp's theme-independent schema.

Story understanding is a related field of research in AI. Limited to the reading of a textual story, the problems inherent in natural language processing mean that the concept of a story is usually reduced to sentence-level semantics, and causal interdependencies in the text [Reeves 2003]. These causal dependencies are usually found by searching for common words or phrases, but there is some new research developing which seeks to use story grammars as templates for understanding text [Elson 2004]. Some story understanding methods have been developed where the story is compared to predefined templates based on story grammars [Norvig web]

3.4. The Crossroads of Interactive Story

Future development in the areas of AI, the simulation of virtual worlds, and interactive story, will be intimately linked. Virtual worlds are the ideal test-bed for human-level AI [Laird 2000], as they remove the limitations concerning sensory input and navigation that robotics faces. An attempt at creating human-level AI should take account of the uniquely human propensity for structuring and storing knowledge in the form of stories, as it is thought to be a fundamental human cognitive activity [Eco 1994]. I will elaborate on this in section 3.4.3.

Currently, one of the most outspoken proponents of interactive storytelling is Chris Crawford. He is convinced that this will be a major area of growth and innovation for both computer games and AI. An early and influential game designer, in 1992 Crawford took a step outside the industry which he saw becoming set in its ways, concentrating his work on developing technologies for enabling the medium of computer games to become as rich a storytelling medium as possible. There was also a group called the Oz project, started in CMU in 1989, that developed some compelling paradigms for istory, notably the separation of the problem into two distinct areas, believable character research and drama management research [Bates 1992]. Brenda Laurel was also an early pioneer, who translated Aristotelian concepts into usable metaphors for human-computer interaction [Laurel 1991]. Janet H. Murray's 'Hamlet on the Holodeck' [Murray 1998] provides a critical analysis of interactive drama, and a set of tools for the analysis of such systems.

There has recently been a surge of interest from academic computer science departments in game AI, along with a general increase in the number of university courses concerning computer games. A number of new projects in istory have emerged, such as [Diaz-Agudo et al 2004]. However, this review will concern projects that have passed the design and development stage and progressed into testing and refinement. John Laird's group in the University of Michigan have recognised the great possibility AI research has in the dynamic creation of story for games [Magerko and Laird 2003]. A recent and impressive istory system is Michael Mateas and Andrew Stern’s Façade system. Façade makes use of Aristotelian concepts, modelling author-level goals and character-level goals in a dynamic real-time system, represented on the screen as a 3-D first-person game environment. Façade will be discussed more fully in section 3.4.3.

The rest of this section will describe some different approaches to AI in istory systems, categorised into four methodological approaches. Mateas and Senger defined three possible general approaches [Mateas and Senger 1999], character-centric, plot-centric, and author-centric, and these are compatible with my distinctions of **story as life** (section 3.4.1), **story as action** (section 3.4.2), **story as thought** (3.4.3), and **story as control** (3.4.4). 'Story as life' refers to both artificial life and natural life, and the view that sees story as an artefact of complex systems. Character-centric istory systems model the characters in a storyworld, and eschew a central control mechanism to facilitate a story that emerges from the player's interaction with the characters. Systems taking this approach will be described under the headings of 'story as life' and 'story as thought'. 'Story as thought' refers to story as a set of relationships between character attitudes and emotions, and 'Story as action' refers to a conception of story as a set of events. Plot-centric istory systems will be described in the 'story as action' and 'story as control' sections, and author-centric approaches will come under the headings of 'story as thought' and 'story as control'. 'Story as control' is a reference to the view of stories as tools that serve an author's purpose.

There are hybrid models that could incorporate all four methods, so the distinction is present in order to show the different levels, akin to Aristotle's levels, at which istory can take place. Some of the systems will later be compared with the OPIATE system, in section 6.2.3. Each of the following sections also relate concepts in istory to ways that stories and storytelling influence human life and culture. This is done because just as AI researchers have traditionally studied biological phenomena to gain inspiration, to create true storytelling AI we must first observe how stories exist in the real world, and how people use them for their many and varied purposes.

3.4.1. Story as Life

For a virtual world to serve the purpose of entertainment, story is **not** always necessary. Some of the most popular entertainment activities would seem to be devoid of story, with sports games being the most popular type of computer game. Yet where a story exists, there is a more meaningful connection with the audience. This is most obviously seen in the movie industry and television industry, although the new trend of 'reality TV' (or as they say in the industry, 'writer-less TV') is letting individuals' real lives take the form of stories under the supervision of editors who endow the lives of the participants with a structure that is translated by the audience into a narrative form.

As was discussed earlier, it is possible to interpret system dynamics as stories. There is more to stories than mere causality, however. Stories are the heritage of ancient man, and reflect ways of thinking that have more to do with fate, doom, and mysticism**,** than the causal dependencies of scientifically engineered systems. The most interesting stories are not lists of simply causally connected events, but events that are connected in coincidental and surprising ways. There are properties of story that do not correspond with a simple recounting of events. Most prominently, a story has a distinct beginning, middle and end [Aristotle 50BC]. Reality is not so constrained, and a simulation system that models believable characters inhabiting a virtual world, such as 'The Sims' or 'Creatures', shares this property. Stories that a player can recount after playing this type of game are usually based on unusual occurrences that are not an obvious side effect of the simulation rules. Although people do tell stories about their everyday lives, an assumption I have made is that these stories are different than those that a computer could be programmed to make based on a list of causal events. People subconsciously structure their stories with the conventional rules and structures that they have picked up since childhood, and it takes an expert story writer who is aware of these unconscious connections to explicitly apply the tools of story authoring to new storyworlds.

Cavazza et al [Cavazza et al 2002] take a character-based approach to istory, modelling characters as agents with plans and goals modelled by Hierarchical Task Networks (HTNs). In this environment, there is no explicit model of plot independent from characters, and the artificial life of the characters is interacted with by the player, who takes a 'mischievous god' role. The author-defined goals of the characters serve as the mechanism for defining the initial state of the storyworld, and the process of interaction consists of 'shouting advice' at the characters, and manipulating objects that are a subject of a character's goals [Cavazza et al 2002b]. Although there are many other believable agent implementations, there are few who claim that character interactions alone create story. Barbara Hayes-Roth is one who rightly views an audience reaction as the intended goal of the story, and distinguishes three feelings that seem to encapsulate this intended reaction: joy, rapture, and enlightenment [Hayes-Roth 1998]. Although she recognises the need for an 'automatic story master' to structure a player's story experience, for her the characters are the most important element of the story.

Figure 3.13: Cavazza et al’s use of the Unreal Engine

The Oz project initially took an approach assuming that two threads of research would enable istory telling on computer, that is, drama management, and believable character agents [Mateas 1997]. Although the emphasis was initially on both components equally, research in the Oz team consisted mostly of believable agent work. A believable agent is characterised by its broad, shallow nature, in that it can act believably, but not necessarily intelligently, in a wide range of situations. Personality and emotion simulation were key concepts [Bates et al 1992b], and the resulting demonstrations captured some extremely effective characters, in cartoon styles [Loyall 1997]. However, the drama management research was not as successful, perhaps owing to the approach. In order to enable plotting in their interactive story systems, it was assumed that the drama manager would have to search all future story possibilities based on the current interactions [Weyhrauch 1997], much like a classical AI planner in an interactive system. This led to the focus on believable agents research and ultimately the ending of the project, which was not revived until Michael Mateas built on the Oz architecture, collaborating with Andrew Stern to create a unified dramatic story engine that incorporated all the personality and emotion models with a drama manager [Mateas 2002]. A description of Mateas and Stern’s work will appear in the next section.

3.4.2. Story as Action

There seems to be a subjective difference between different people's idea of a story. Some people see a story only existing in the dialogue of characters, and some think of story as the events and actions that occur, with dialogue being ancillary. This reveals a general difference between two distinct types of story. Soap operas and dramas are based firmly in depictions of realistic characters, yet always contain a plot that structures the characters' experiences. The audiences of these stories see the plot as a product of the characters' conflicts, instead of the Aristotelian reverse. The author expresses themes through the analysis of each character as a realistic depiction of an individual, through dialogue, and emotional resonance's between the audience and the characters.

The action and fantasy genres in movies and novels often deal with the plot as the central focus of the story, and the characters are often only fleshed out enough to serve the plot. The author retains a distinct presence, as realism is sacrificed for the author's vision, and the events in the story implicitly define the themes. The actions of the characters are required to serve the plot, and secondarily inform the audience of the characters' qualities. A lot of modern stories mix elements from both types of approach. A piece of dialogue or action can be effective at both unfolding the plot and for character development, and when it does both, a certain harmony is created [Andrews 1915].

What sets Propp’s work apart is the fact that it is independent of verbal expression. Aristotle initially notes that the plot, as a series of actions, is the most important part of a story, but goes on to show the importance of words and how they are used in his examples. A Proppian story is carried out primarily through actions, not words. He mentions motivations and connective elements but goes into little detail. This is not because he saw them as less important to the story, but the opposite. Elements such as emotional motivations and descriptions of appearance are at the core of storytelling, but did not have primary importance in Propp's morphology. These elements are specific to each story's internal world and work according to the rules that an author sets up, different for each one. The expression of specific themes in a story is dependent on them. The morphology of plot works at a higher level, to structure the events that occur in the story-world so that they are intelligible to an audience in relation to a universal conception of the hero's journey.

It has been postulated that Propp's morphology is applicable not just to the Russian folktales from which it was derived, but to myth and folktales from cultures around the globe. It has been used to analyse the Bible [Milne 1988], Vietnamese folktales [Vu Kim Chinh 1997], Native American folklore [Dundes 1964] and contemporary movie stories. This shows that although the individual works can evoke many themes and emotions, they share a unique structure independent of these. Dundes has shown [Dundes 1982] that Propp's schema does not fit some folklore because some, such as the Grimms, have taken original oral tales and mutated them, injecting elements from literature, and therefore these tales can no longer be classified as folktales. In recent years, Propp's work has been incorporated into a number of computer systems, with different results. Apart from Dundes's early implementation, current work by Braun [Braun 2003] and Grasbon [Grasbon 2001], Paiva's group [Machado et al 2001], and Magerko and Laird [Magerko 2003] show the diversity of products that can be enabled with a Propp-based conception of story.

Norbert Braun has completed a number of different implementations of istory systems, and each is informed by Propp's work. In Geist [Braun 2002], an augmented reality(AR) installation at Heidelberg castle, the user dons a headset and views the world with an overlay screen, where virtual ghosts are projected, in order to experience a guided tour around the history of the area. The user can walk freely around the grounds, interacting with the ghost characters, and the system automatically generates a plot based on pre-scripted narrative segments that can be experienced in different orders. Braun uses Propp's own algorithmic approach to creating a new plot out of character functions:

‘‘It is possible to artificially create new plots of an unlimited number. All of these plots will reflect a basic scheme, while they themselves may not resemble one another. In order to create a folktale artificially, one may take any A, one of the possible B’s, then a C, followed by absolutely any D, then an E, then one of the possible F ’s, then any G, and so on. Here, any elements may be dropped (apart, actually, from A or a), or repeated three times, or repeated in various aspects''[Propp 1968].

In Propp's terminology, A denotes villainy, B mediation, C counteraction, D the donor function, E reaction to the donor, and F the provision function. These character functions concern the villainy, the hero's reaction to it, and the means of acquisition of a magic object for use in the quest, and will be elaborated in the Design section. Braun's implementations involve taking a setting and group of characters, and categorising pre-defined events as certain types of character function. These functions are carried out in Braun's system as the execution of the currently active event, with any user interaction or movement contributing to the decision of the next active function. User interactions in the system are limited, and future work on the system will be involved with allowing user participation as conversations, using natural language parsing [Braun 2002b].

Grasbon's work with Braun's system has consisted of developing ways to incorporate user interaction into a dynamic unfolding of the plot [Grasbon 2002]. He has developed an architecture using the 'polymorphic function', a function requiring the user's reaction, as key points in the tale that can react to the user through repetition of functions. These functions were annotated by Propp as positive or negative, for example, the function E+ means that the reaction to the donor is positive, resulting in the F+ provision function, while E- denotes a negative reaction, meaning the D donor function is repeated, if necessary with a new Donor character (see Figure 3.14 for a graph).

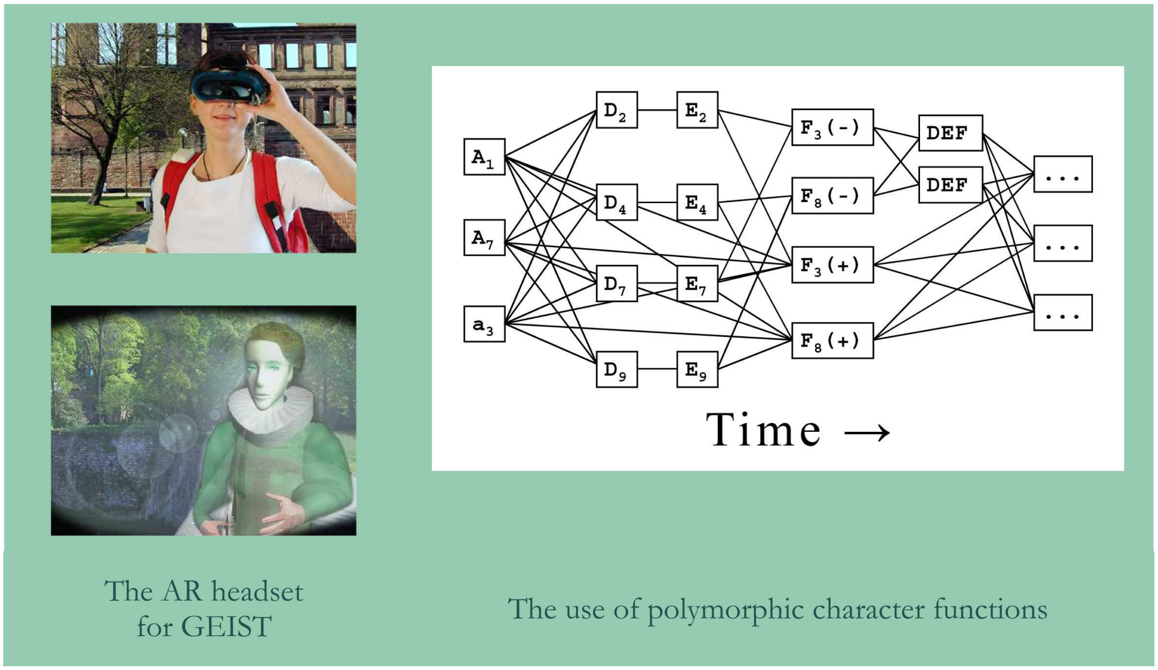
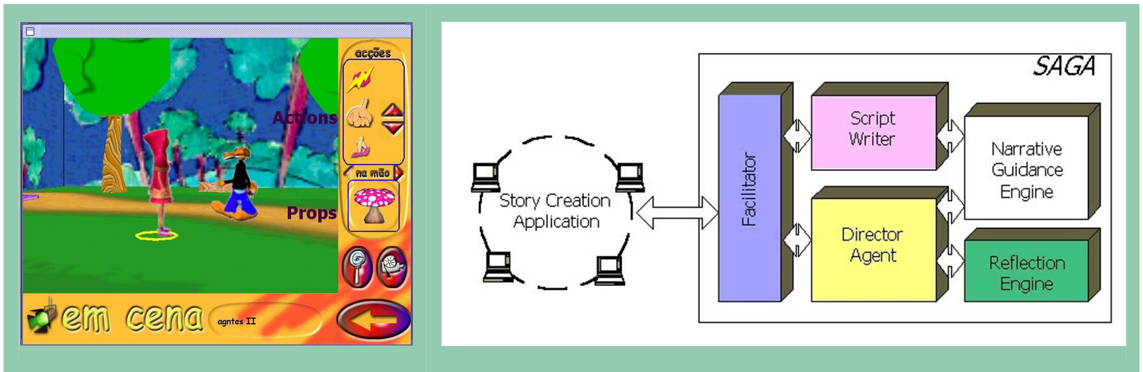


Figure 3.14: Braun's GEIST Implementation

Ana Paiva, Isabel Machado, and Rui Prada used the Propp functions to enable a children's storytelling tool called TEATRIX [Prada et al 2000], were each player takes control of one of the characters, and the system allows players to chose pre-defined scenes to construct a story in real-time. The children's choice of character effects the abilities that they have in the storyworld, and the Proppian roles that they will play are chosen before the story begins.

The direction of the story is done by another player, who takes the role of story 'owner'. In later implementations, an augmentation to the system involves the integration of the SAGA (Support And Guidance Architecture) [Machado et al 2001], which dynamically suggests new events to the children, and can introduce new objects or characters for manipulation. This incorporates a Director agent which is able to recognise Propp functions in the players' interactions, and reacts by providing ways for the situations to be resolved. TEATRIX is a novel approach to applying Propp, and is the first system which explicitly integrates character functions into a multiplayer environment.



TEATRIX

Integration of the SAGA system

Figure 3.15: Teatrix

Magerko and Laird use an extension to the Unreal Tournament game and model a story as a pre-scripted plot, with no possibility for the user's interactions to effect the outcome [Magerko and Laird 2003]. This approach was justified as a perceived balance between a writer's vision and a player's interaction. The player's interaction is based on the concept of Weyrauch's temporal variability [Weyhrauch 1997]. The story is told as a series of scripted NPC actions, for example in a ghost story in "Haunt II", and each action must occur, but at times that depend on where the player is situated in the environment. Each action occurs at a particular state of the story, and the SOAR rule-based architecture is used to decide when actions should happen. The limitations of this approach mean that true player agency is not enabled, and the freedom inherent in free-roaming computer games is not possible using this approach.

A related project, also using the Unreal engine, is MIMESIS, headed by R. Michael Young. Interactive plotting is conceptualised as a planning problem, with the system reacting to dynamics [Young 2000], and player interaction, by means of either intervention tactics or re-planning. The planner initially makes sure that any actions the player takes do not interfere with the plan, instead of incorporating this information into the creation of a new plan. This is done by intervention, which entails preventing the player from succeeding in the action they wish to make that may interfere with the eventual outcome of the plot [Riedl et al 2003]. An example is seen in the bank robbery demonstration, in which the player can apparently shoot the bank robber before he gets the chance to rob the bank. If the player attempts this, the system will automatically force the player to miss, as it conflicts with the planned bank robbery event. If player interaction does not conflict with the current plan, then the plan goes ahead as normal. Although this model allows for player agency to some degree, it does not allow for player agency to effect the plot. This is a problem for an istory system, but is justified as a balance between a writer's vision and the player's interaction. A complex re-planning scheme is also part of the Mimesis system to use as a backup, although this does not happen in real-time, but is done in advance.

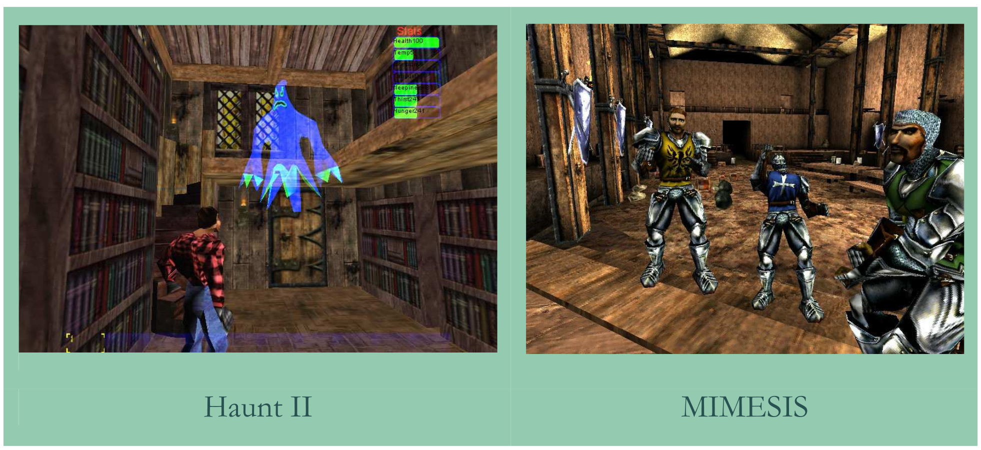


Figure 3.16: Istory systems using the Unreal engine

Chris Crawford's Erasmotron engine was an ambitious project, which takes an original approach to istory, in that it attempts to incorporate a wide range of possible models of story, and implement them in one system. Characters are modelled as complex systems of personality traits, emotions, attitudes, and acquired knowledge. The interactive plot is modelled as a network of possible sub-story fragments, each of which can apply to different characters, and each of which have a number of possible reactions, leading onto other fragments in the network. This system cannot be classified under one heading alone, yet because actions form an integral part of the system, this component of the Erasmotron engine will be discussed here. Crawford saw the use of a wide range of interactions as the key to his system's success, and created hundreds of verb descriptors, classified into about thirty categories, from verbs about sex to verbs about attacking, and everything in between. Because so many verbs were available, an author of an Erasmotron storyworld could realistically use only a subset of them, having to define all the characters' personalities, along with substory fragments that incorporate the verbs. The availability of so much choice in interaction means that the player experiences the feeling of inhabiting a much richer storyworld. Crawford's experience shows that the more complicated an istory technology gets, the more care will have to go into authoring storyworlds for it.

3.4.3. Story as Thought

Not to be confused with Aristotle's layer of thought, which signifies everything that is asserted in a story, I will here discuss the implications of some research into narrative intelligence. Narrative intelligence is about the role that stories have in the evolution of human thought, along with istory systems that attempt to model this. When classical AI was first being researched, and leading scientific minds believed reason to be the highest form of human thought, the algorithms that were developed were based on the solution of intractable puzzles and deep search problems. As AI research developed, people found that the most difficult problems in creating AI were to be found in less traditionally reasoning-based tasks. Computers can be programmed with ease to do things that 19th century philosophers and scientists held to be the epitome of human achievement, such as mathematics and logic. Consequently, the role of emotional intelligence, social intelligence, creativity, and inspiration, has become stronger in AI research. These, like with vision and pattern recognition, are areas where humans excel easily, yet it is very complex to develop an AI system to model them.

Current scientific research is considering the model of stories as a fundamental means that people use to structure thought [Turner 1996][Eco 1994]. As part of the 'Discworld' series, in 'The Science of the Discworld II', Terry Pratchett with Cohen and Stewart [Pratchett et al 2002] discussed the role of 'narrativium' in shaping humanity's cultures. This book was a 'popular science' book, for the layman, yet touches on some currently accepted scientific theories. Narrativium was proposed as a fictitious element that exists only in people's minds, actively promoting imagination and the development of ideas. Narrativium is nurtured by the stories that people disseminate, and it controls how the world evolves, indirectly, through the actions of people. The stories in people's heads give them familiar scenarios that they can compare to the events in their lives, and then they can act based on the second-hand experience that the stories endow them with. People grow up with their parents telling them fairy tale stories, and this is seen as a vital part of a Pratchett's 'make a human kit' (from [ibid]) which passes on unconscious knowledge, or knowledge that we have no other known way of passing on. Thus, the causal, temporal, and coincidental structures contained within stories become part of the basic mechanisms of human thought.

I would argue that Language is a basic structuring mechanism of human thought, affording sophisticated communication, and linguistic structuralism implies that linguistic relationships are also a source of human conceptual relationships [Gopnik and Choi 1990]. Children learn to use language, starting with rudimentary vowel sounds, vying to be understood, until they can express themselves to their parents well enough to ask for things and explanations. Maria Montessori saw that a child's mind is far better suited to learning than an adult's, with a natural curiosity that enables them to readily infer meanings and associations through contact with a combination of language and sensorial experience [Montessori 1966]. When storytelling is introduced, the child begins to develop a longer attention span, being able to construct the meaning of a series of incidents told in temporal sequence. Even if the child infers the 'wrong', or unintended, meaning of the story, the process of forming a meaning for the story helps build the child's abilities as a thinking being.

The last paragraph is my own argument advocating the importance of storytelling in structuring human thought processes. The process of constructing a meaning out of a longer (than a sentence) verbal structure is mental preparation for constructing our own meanings for real-world events, allowing sequential reasoning skills to be developed.

The istory approach of using a virtual narrator to present the story has produced a number of successful systems. Braun applies this approach in the Geist system, where stories are narrated by the virtual ghosts that users encounter. Szilas [Szilas 2002] is firmly in the plot-based camp, using some structuralist extensions to Propp's work in modelling the thought processes of a narrator telling a story. Claude Bremond was not satisfied with Propp's character function as the basic element of the plot, and identified several processes that occur at the plot level [Bremond 1974], and identified binary oppositions between distinct actants. Using this model, Szilas has attempted to simulate the process of narration, to facilitate truly interactive narrative. However, because he uses structuralist models of reduced complexity, the plot is represented as a series of obstacles that characters must overcome [Szilas 2003], ignoring the more empirically based, and complex, model of Propp. This is reminiscent of Cavazza, and the earlier Meehan's, approach, yet instead of simulating characters, it incorporates a more plot-based simulation model, which allows for user interaction in the unfolding of plot. The text-based approach is limited, however, and Szilas informs us that future work will be in more graphical environments [Szilas et al 2003]. Szilas' approach is conducive to interactive plotting, and he notes that the primary requirement is that characters behave according to the dictates of the plot rather than models of their own motivations.

Sgouros uses an Aristotelian plot model in his system, which is independent of environment, and has implemented the 3D DEFACTO game to showcase it [Sgouros 1999]. The automatic resolution of the plot is an integral part of his system, and proceeds by analysing character states and previous actions. Aristotle's dramatic arc is central to Sgouros's plot simulation, and the player's interactions trigger complication events, that serve to increase the level of conflict in the story, and resolution events reduce the level of conflict until a conclusion is reached. Characters are not autonomous, and serve the requirements of the plot. The author defines the initial situation, and the system simulates the author's thought processes in bringing the story to a satisfying conclusion, via conflicts that emerge based on the player's actions. The main limitation I would see in this is that the plot models are too simple, made up of simple conflict/resolution pairs that would not allow any higher-level plot control by an author.

Mateas has greatly contributed to the development of istory, and his 'Façade' game was a finalist in the respected Independent Games Competition in 2004, and also appeared in popular press such as Time and Wired. Mateas is one of the most oft-cited istory developers, with his discussions of narrative intelligence [Mateas and Sengers 1998], and neo-Aristotelian drama [Mateas 2001]. Façade features NPCs with simulated emotions and personality, which the player interacts with by taking the role of another character, using a unique natural language engine [Mateas and Stern 2004]. The system is built using the Oz architecture, using a drama manager to direct the actions of the NPCs, who are implemented as semi-autonomous. Mateas defines a scale from 0 to 100 [Mateas 2002], where 0 is fully autonomous characters, and 100 is where all character actions are initiated by the drama manager. Façade falls at around 70. Low-level reactions to the environment are carried out by the characters, but any event that has any relation to the story is controlled by the drama manager. The most important story concept that Mateas uses is the 'dramatic beat' - the smallest event that changes the values of a character. These dramatic beats are planned and scheduled in a similar way to Sgouros's system, calculated to create and then resolve conflicts between NPCs, and between NPCs and the player character.

Crawford also models characters in a complex way, and the development of the Erasmotron story engine consisted largely of defining all the possible categories of emotion, personality, and actions that characters could exhibit. Plots are authored as networks of abstract segments, with the dialogue pre-authored as discrete character-specific chunks, which can be reused in different contexts. When a plot segment is reached, it is executed by utilising the current state of the relevant character, and the type of action that the plot fragment requires, to create a dialogue event that is presented to the player as text, alongside a depiction of the character's emotional state in an image. The plot fragments are pre-authored, and a player influences the path of the story through the plot by choosing from a list of available responses. Although the plot network is pre-authored, each plot fragment is an abstract representation that can be enacted differently depending on the state of the character that enacts it. Crawford also innovated in is use of gossip simulation. First attempted in his game 'Trust and Betrayal', the Erasmotron incorporates a sophisticated procedural model of gossiping [Crawford web].

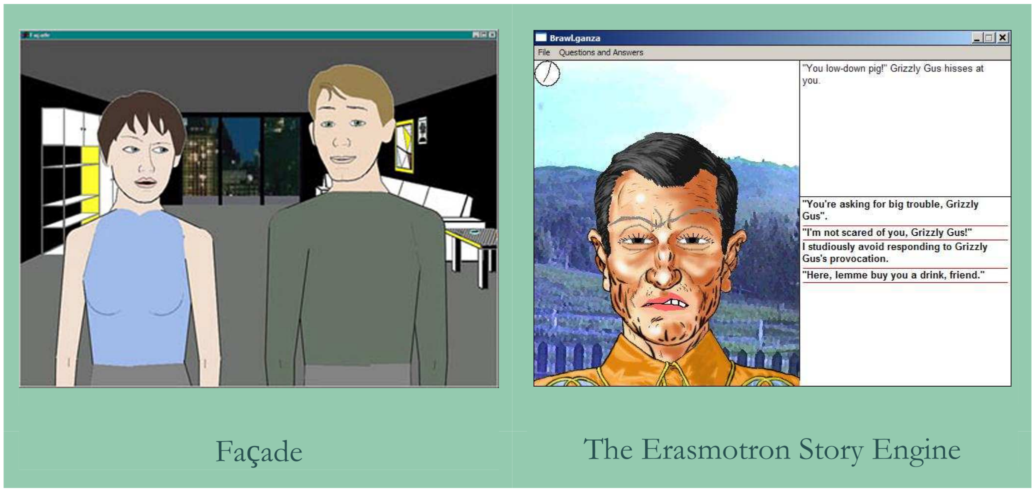


Figure 3.17: Mateas and Stern’s and Crawford's Story Systems

Another contribution Terry Pratchett made was explained in [Abbott 2002], which references the novel 'Wyrd Sisters'. In this book, set on the magically infused Discworld, the narrative drive which usually structures events in Discworld goes out of control and is anthropomorphised as the villain of the story. The heroine, Nanny Ogg, strives to counter the onward push of the narrative by going against every clichéd occurrence, never giving in to the obvious requirements of the plot, until she gets to the bottom of what is causing this - the overabundance of narrativium. Pratchett here provides a compelling metaphor for players in an interactive storyworld, which is related to the experience of players in story-based computer games. It has been casually observed that players instinctively know that in order to advance the plot they will have to venture into new territory, but often put this off by exploring every alternative before opening 'that door', which they know will lead on to the next challenge or story point. This resistance to being prey to a set-in-stone narrative is a fundamental reason why consumers seem to be increasingly turning away from their television sets (as observed by Ziff Davis Media group [Ziff web]) to experience the greater freedom in, and control over, the experiences provided by computer games.

3.4.4. Story as Control

The use of storytelling to achieve control over people seems quite eerie, but has been fundamental in humanity's history. This was a theme in 'Science of the Discworld II', mentioned in the last section. In some works [Propp 1984, Lang 1996], it is proposed that stories grew out of prehistoric rituals, particularly initiation and death rites. The specific motifs used in stories come from ever-changing cultural, social, and technological sources, yet it can be seen that structures and plots in modern stories can be traced back to ancient versions. For example, the plots and conflicts found in comic book superhero tales can be traced back to mythologies from ancient civilisations [Johnson web]. The myths and legends of today represent belief structures of previous civilisations, and served to keep people from acting outside social limits. Northrop Frye's conception of myth as an expression of a societies' concern, that works to balance freedoms [Frye 1957], is a corroboration of this idea.

When an author expresses central themes with a story, they usually do so by putting forward a theme along with a judgment on that theme that is illustrated by the ending of the story. Star Trek's plots are moral fables [Barad and Robertson 2000], that put a dilemma to the hero (most often the captain of the starship) and show how a moral judgement results in the successful return to a state of equilibrium. The control exercised here is an antithesis to the freedom found in a simulated interactive world like a computer game, yet the process of learning from the interaction can also serve to elucidate themes in a different way. Although the traditional happy ending shows how a moral judgement results in a return to equilibrium, there are other forms that take a different thematic approach. Martin Scorcese's 'Taxi Driver' features the anti-hero of Travis Bickle who takes the law into his own hands, acting on paranoid delusions, yet he receives the happy ending that is usually reserved for the hero that takes the moral high ground. Scorcese's storyworld here serves as a mirror on reality, where the good guy doesn't always win.

The use of simulation to express artistic themes is a method that was used in Chris Crawford's computer game, 'Balance of Power'. In this game, the player takes overall strategic control of either the USA or the USSR during the time of the cold war. It has been observed by Ernest Adams [Adams 2001] that in controlling the USSR, the player is faced with the perspective of all the surrounding allied countries being 'bought off' with the USA's economic superiority, and the player controlling the USSR has to manage its resources far more efficiently in order to compete. This was a revelation that reminded Adams of the dawning of awareness that often accompanies the experience of an artistic work. Picasso's cubism, based on the method of depicting a subject from many points of view simultaneously, is a paradigm that has parallels in interactive computer games. If the player is allowed to choose a variety of points of view (different characters or personalities) and perceives the game world differently from each, both the player and the author share control of the experience. This sharing of control allows a flexible artistic expression, which has more in common with a conversation than the one-way exchange usually associated with artistic expression, in painting or writing.

The author of such a flexible game world must anticipate any choices the player can take, *unless* the world is simulated to a degree of complexity that allows consistent thematic expression independent of explicitly authored content. It can be argued that 'The Sims' constitutes this sort of thematic simulation, and Will Wright has admitted that he believes people can come away from the game with a new sense of the need for balance in their lives. Just as the balance between pleasure and duty rules the lives of the 'Sims', balance is a vital part of living a daily life, and people become aware of this 'balancing act' that they are subconsciously playing with their real lives as a result of playing the game [Twist web]. Controlling the expression of theme through simulation is a complicated task, which requires skill in a broad range of new fields, therefore it is not yet a recognised art form. Musgrave is a champion of procedural artistic expression on computer [Musgrave 1995], and identifies the practice as one that will perhaps never become popularly accepted as art, due to its dependence on a wide range of skills, both technical and aesthetic.

The simulation of narrative intelligence, seen in the istory systems described earlier in this chapter, is an attempt to model the process an author goes through in implicitly elucidating a theme with a series of events. Because Propp based his analysis on Russian folklore, we see in his morphology an abstracted expression of the *themes of Russian society* rendered as a generic story template. The causal relationships between events are secondary to the temporal nature of Propp's scheme, yet because the ancient societies that inspired folklore saw events as being connected in a different way than we do (exemplified by the ancient Chinese I Ching), the temporal structure could in fact be a model of the way these societies viewed causality, much like Jung's idea of Synchronicity [Jung 1953]. The choice to use Propp's structures for OPIATE was based on an assumption that the structures of these folktales themselves contain thematic knowledge that is not expressed in Propp's description of them. The thematic power of these structures comes from the fact that they endured so well as oral tales, resonating with enough people to survive in the memories of generations.

Thematic author modelling is a concept used to great success by Turner for the MINSTREL system [Turner 1992], yet has few modern proponents. The BRUTUS system, by Bringsjord and Ferucci [Bringsjord and Ferucci 1990], uses an author model to exclusively represent the theme of deception in the events depicted by the story. The characters could use deception to achieve goals, and because an extensive model of the use of deception was integrated into the system, various complicated plots could emerge. However, because this thematic knowledge was designed into the system in a top-down fashion, BRUTUS was limited in the themes it could use, and integration of use of other themes was deemed too unwieldy to pursue. A more bottom-up simulation that incorporates thematic mechanics could allow for emergent occurrence of thematically interesting events. This is unknown territory for istory developers, yet Turner seems to have made considerable progress with his static textual stories, and the moral themes in MINSTREL's output are based on simple author-defined templates that can be abstracted for use in many contexts. There are many lessons to be learnt from both his, and all the other systems described in this chapter.

3.5. Conclusions

To link some of the background provided in chapter 2 with the current chapter, I would like to make the point that as well as providing a good test-bed for AI research, computer games are a venue where Jung's collective unconcious and Levi-Strauss' structuralist myth-mind have a very real applicability, regardless of how accurately they seem to model reality. These concepts enable a structure of indirect control over a virtual world that is not embodied in that world, but is implicit in the definition of the system.

In section 3.2, I show various means and styles game designers use to portray stories in their chosen medium, and give an introduction to the use of simulation techniques for creating interactive systems with emergent gameplay. In section 3.3, I introduce some AI research that deals with creativity and show how the most useful systems in this area are those which combine elements of different AI architectures. I also introduce artificial immune systems as a good example of a system of different evolving entities which also incorporate a mechanism for global memory, influencing how the system evolves.

In section 3.4, a number of systems are dicussed that have similar goals to the OPIATE system in order to provide an idea of what problems are faced in creating an istory system, and some different approaches to overcoming them. The rest of this thesis is a description of my own approach. This approach is the combination of a character-based approach with a plot-based approach in a way where the workings of both mutually benefit the other. This is achieved by using a Story Director agent as a global director which keeps track of the NPCs in the system as a whole, while being focused on the activities of the player character. This architecture allows the indirect control implied by allowing a certain amount of autonomy in the NPCs while influencing their goals. This could be seen as a model of an 'unconscious drive' for the NPCs, and allows the NPC dynamics to form a complex system where the emergence of social situations between the characters can occur. The Story Director (SD) agent then represents a common driving force with a shared memory of stories.

4. Design

4.1. Introduction

The primary design goal of the project is to create a computer game system where the player has a maximum of potential choices at any one time but is also constantly engaged in a consistent story that is not explicitly authored by a person, but dynamically generated in real-time. The approach is to develop a story management system which monitors the events in the game world, creates appropriate story plots for execution in the game world, and directs this story as it is played out in the game. The storytelling techniques found in certain commercial games are assumed to be valid and robust, as proven by their popularity, although the plot in these games in not interactive.

The design approach entails the separation of the architecture into (I) the game engine, and (II) the Story Director (SD) agent. The SD is an agent that operates in the environment of a game engine, directing the non-player characters (NPCs) therein to enact a story for the user. A *game engine* is a game industry term describing a set of tools and resources that can be reused in different games, and consists of different components such as a graphics engine, and a physics engine. A design decision was made to situate OPIATE in a 3D game similar to commercial games, and to use some storytelling techniques from them.

The 'Unreal Tournament' game engine is freely available, and easy to customise, thus it has been used for a number of istory projects such as MIMESIS. This game engine was evaluated for use in this project, and, although it was an attractive option with most game engine-specific problems already taken care of, it was found to be; (a) not flexible enough as regards the use of story direction methods, and (b) too focused on violent character interactions. Therefore, a new game engine was developed, to maximise flexibility and allow any features that were desired to be implemented (and to gain experience in the process of game development).

The overall architecture of the design was based on the Oz project's division into believable characters and drama management [Mateas 1997], shown in Figure 4.1. This also corresponds with Propp's assertions that characters, motivations, and themes are independent of his plot model, this correlation being an early motivation to the development of the system. The characters of a story world are author-defined in (a) how they look, and (b) their personality and history. Their motivations are based on (b), along with the events that take place during the course of the story. In general, themes can be expressed in many ways, but the design of the OPIATE system proceeded with no real mechanism for allowing an explicit definition of theme. The design of the world, the characters, and the possible interactions was assumed to be a method for implying themes. Thematic interactive plotting (making the story be *about* something) is thus a difficult problem, as the organic unity of the story has been split, handling the characters and the plot independently. However, although Propp maintained that while the plot and characters are conceptually independent, in specific stories they are united and correspond in thematic ways, specific to each storyworld. Aristotelian **patterns** are to be seen here, and serve to unite the structural plot skeleton with the characterisation to form a consistent thematic purpose.

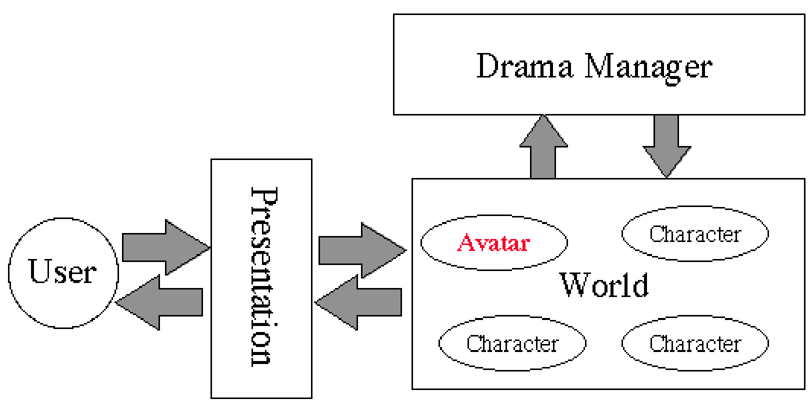


Figure 4.1: The Oz Architecture

The design of the system at a lower level was carried out as implementation proceeded, using what is known in some game development circles as an *organic design process*, or evolutionary prototyping in more academic circles. This chapter will begin with a discussion of the design goals of both the game engine and the story director, and will then move on to how these requirements were addressed on each of Aristotle's layers. This section is presented to illustrate the design of how a story is to be told with OPIATE, and serves as a functional specification of the system. The chapter will then move on to describe how the system was designed to facilitate the functional specification in the next sections, discussing the software architecture, and then detailing the two most important architectural components, the game engine, and the story director (SD) agent.

4.2. Requirements

One requirement for the system is that it should deal with the plotting at an abstract level, so that the system can be applied to different storyworlds, and so that the approach and architecture is valid for other game engines. The SD agent should operate at a certain conceptual distance from the game world. Although it was not deemed necessary to have a separate software process running, the Story Director Agent was intended to be as loosely coupled with the game world as possible.

4.2.1. Story Director

The SD was envisioned as a software agent, using a definition of agent that requires (a) a sensory knowledge of the world in which it operates, (b) the ability to create plans to change that world based on its own knowledge and drives, (c) the ability to effectively carry out these plans, and (d) an ability to react dynamically to the world and change or adapt its plans.

The story is to be initiated by the SD, based on an expert knowledge of the skeletal plot structures of Propp, and how they fit into storyworlds. These story plots are to be enacted based on the dynamics of the gameworld, and must react to player interactions. It was decided that the SD should operate in a way that enabled open-ended stories. There is not a desired ending that is forced, but the story must progress in a satisfying way and be seen to make sense. Because of the open-ended model, the player can influence the plot to a considerable degree.

Any artistic themes in the plot are assumed to be implicit in the skeletal plot templates, but themes are more explicitly incorporated at the game design level. The ability of an author to effect the story is limited because the system allows an open-ended story to emerge. The author defines the choice of plot models available, and the interactions that characters can have. For an extension of the system that allows better control of the SD agent, an author interface would be desirable which can adjust the parameters that dictate the details of how the story is planned and enacted.

4.2.2. Game Engine

The game engine was designed as an adventure game engine. The three main components in standard graphical adventure games, such as the Lucasarts games, are characters, objects, and locations. Each of these components are created by the game designer, yet basic attributes are coded into the engine. The player takes control of a character and travels between locations, acquiring and using objects and interacting with other characters to advance their progress through the locations that make up the game world. An important requirement of OPIATE's game engine, that is distinct from other adventure games, was that the NPCs should be capable of moving around locations like the player character can, and act within a social simulation that includes the player character. This is necessary for the OPIATE system, as the abstract plot-level planning can only be truly useful if it can be applied to a dynamic system where the skeletal plot events can be fleshed out with characters that respond to the player.

Because of the open-ended plotting, the game environment must be created with a view to defining the possible paths of progress through the game. Because of the choice of using an adventure game as a testbed, puzzles involving objects are important. However, since the SD plotting must be compatible with other types of games (e.g. Action games and RPGs), these puzzles are ancillary to the planning of the plot. This illustrates a limitation of the game engine. A typical commercial game incorporates many challenges and action elements, yet in testing the OPIATE system, storytelling capabilities are specifically being examined, so designing a challenging game was not a primary requirement. If the OPIATE system was to be incorporated into another game, many other elements would be necessary, but the elements deemed important for interaction with the game world were as follows:

- Character modelling that incorporates attitudes, behaviours, goals, and an inventory of objects.
- Objects that enable new actions to be taken, and portal objects that link locations.
- A gossip system which enables knowledge of story events to be disseminated through the characters.
- A dialogue generation system incorporated into the characters. This does not need to be sophisticated natural language generation, but needs to be able to get across the simple communication necessary for the player to be aware of story and gossip events.

4.3. Layers

There were two distinct architectural levels of design involved in the OPIATE system; the game world and the story director. However, in describing the design of the system as a storytelling medium, I will here refer to Aristotle's layered description of stories, from the layer of plot on down to spectacle.

The design of each level's functioning was planned to work together with the others, and the flow of information is almost always from a higher level to a lower. The case where a lower level informs a higher level is that of character to plot. This is the difference that the interactivity of the system necessitates. In Brenda Laurel's 'Computers as Theatre' [Laurel 1991], Aristotle's layers are depicted in the context of human-computer interaction, and she puts forward the idea that in an interactive system, the lower layers must inform the higher layers, but here, this need only be true of the character/plot interaction, as the player controls the interactions of the hero character. Figure 4.2 shows how Aristotle's layers correspond to levels of information content in OPIATE.

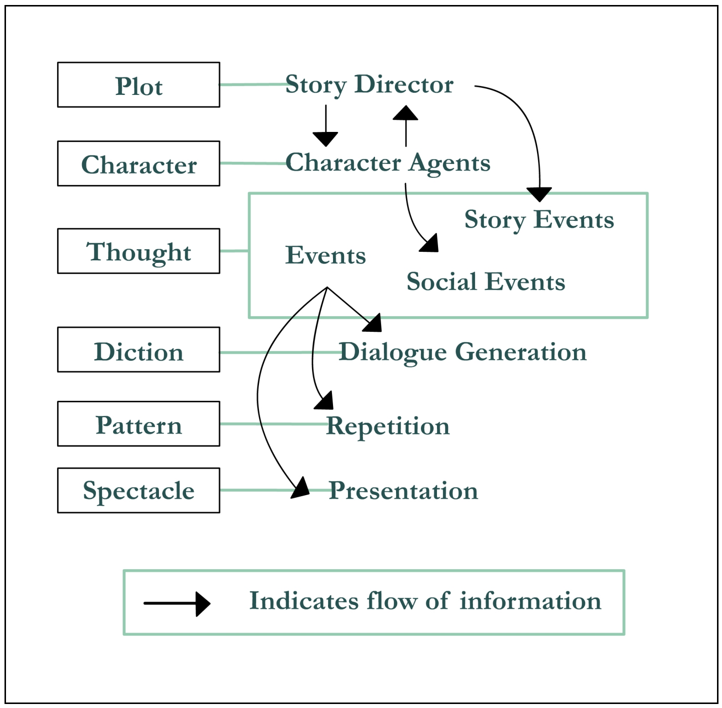


Figure 4.2: Information flow between Aristotelian layers in the OPIATE system

Edgar Allen Poe did an extremely methodical analysis of the creative process that went into his poem ‘The Raven’ [Poe 1850], in which he maintains that the creation of a work of art necessarily begins at the end, or purpose, of the work. The purpose of the poem was to appeal equally to critics and a wider audience simultaneously, and based on this purpose he developed a theme, tone, structure, and a focus, which was the word ‘nevermore’. This base was then extended to define the choice of rhythm and rhyme schemes, and the character motivations, and finally the images and events that are present in the work. His analysis is a great insight, yet does not directly enable systemation of the processes behind the work. Constructing an AI to perform these generalised tasks is impractical, requiring too much knowledge and metaknowledge, along with an intuition for the human condition. His layered approach is useful, as, like Aristotle’s, it could inspire the creation of an author interface for interactive storytelling. The top level of a narrative, encompassing theme, tone, and overall structure could be controlled by an author, allowing an AI to extrapolate a plot's finer elements while incorporating user interaction, within a simulated world. The hierarchical arrangement of storytelling elements in the OPIATE system is shown in the following sections, begginning with a discussion of plot.

4.3.1. Plot

The plot is modelled as a series of character functions, as defined by Propp. The SD agent's only capability to change the storyworld is in its use of these character functions. Propp's definition of the character function is one of his most important contributions to folklore studies, and a thorough analysis of his corpus of hundreds of Russian folktales led to his discovery of only 31 distinct functions that a character can take. Character functions are defined as *'An act of character, defined from the point of view of its significance for the course of the action'*. It is important to note that character actions can appear to be the same, but since they are defined from the point of view of their effect on the plot as a whole, similar actions can in fact represent completely different functions.

A list of all the 31 character functions as defined by Vladimir Propp, along with the symbols Propp assigns them (β, A, B, Pr, etc.), along with how these functions have been interpreted for use in OPIATE, is presented below in Figure 4.3. Each function can be omitted, with the exception of either the Villainy or lack function. However, when included, functions must appear in the following temporal order. Some functions can also be tripled (repeated three times). This list is fundamental to the operation of the system. The zeroth function, which is the introduction of the setting (symbol α), is omitted from Propp's scheme, and can foreshadow other elements, but does not serve to push the plot forward. The functions are split into 5 sections; **Preparation**, **Complication**, **Transference**, **Struggle**, and **Return**. There are dependencies between functions. Dependencies are structured such that if the first one happens, then the second one should also happen, although the second ones can sometimes exist independently. These function groupings are further discussed in section 4.6.2.

Section I: Preparation		
1. Absentation	β	The characters introduced at the start of the story, close to the Hero, can be called Family in this scheme. Absentation involves one of these characters setting out for some reason. This can be done by giving a character a goal to follow whose object is in a different location to the initial one.
2. Interdiction	γ	When the Hero is warned not to do something, or ordered to do something. The order is carried out by a Family character. If the Absentation function was present, the Interdiction involves it.
3. Violation	δ	The Hero violates the warning, or fulfils the order. This act can enable the Villain's reconnaissance by, for example, alerting his attention.
4. Reconnaissance	ε	The Villain, or an underling, looks for some precious object or character to perform villainy on, to form his evil plan.
5. Delivery	ζ	The Villain appropriately finds what he was looking for.
6. Trickery	η	The Villain attempts deception to prepare for the act of villainy.
7. Complicity	θ	The Hero or Family member succumbs to the trickery.
Section II: Complication		
8. Villainy	A	The act of villainy can be abduction, theft, ruining of some object, murder, declaration of war, etc. This is dependent on the resources of the game world, and is chosen based on the dynamics of the world.
8a. Lack	a	Instead of the villainy, a lack can be expounded that does not have a villain behind it. An NPC requires some object or deed to be accomplished.
9. Mediation	B	The villainy or lack is made known to the Hero. A witness character is given the goal to tell the hero of the act of villainy.

10. Counteraction	C	The Hero decides on counteraction. As the player must decide on this himself, counteraction is an event that the SD is not responsible for. The mediation event can be repeated, or even the villainy event can be repeated until the player decides to counteract. The decision to counteract must be explicitly declared or signified by the Hero. If the player does not react, a new plot can be introduced. How these decisions are made will be elaborated in section 4.6.4.
Section III: Transference		
11. Departure	↑	This is the end of the set-up part of the story, and in a game world, this is when new areas are opened for player access.
12. Test	D	A character selected as a Donor is assigned the goal of testing the Hero. This function can take many forms, such as a quest, a battle, an offer of employment, a request from an injured party, etc., and hence is very dependent on the Donor character, the game world state, and the player character.
13. Reaction	E	The Hero's reaction is ideally down to the player, and the test function can be tripled to get a reaction from the hero. The player must be given the chance to achieve the receipt of the reward through any means they see fit.
14. Receipt	F	The receipt of a magical(useful) object. The object will later help the Hero overcome the Villain in the struggle or task functions. The hero can receive the service of a Helper character instead of a useful object.
15. Guidance	G	This function is another function, like departure, which involves the transportation of the Hero and the opening of new areas, this time to the locality of the Villain. This can be done with a flying horse, a train, by opening a gate for the hero, or some other means specific to the gameworld. The test-reaction-receipt section is often repeated or tripled before the guidance function
Section IV: Struggle		
16. Struggle	H	This struggle is with the Villain, and depends on the game type and player interface. It could be a riddle, a test of skill, or a competition. In my game engine, the struggle usually takes the form of an object puzzle that must be completed.

17. Branding	J	During the struggle with the Villain, the Hero can be branded with a scar or given some means by which he can be recognised. This function works with the later Recognition function. This could be a graphical change in the appearance of the Hero.
18. Victory	I	The Villain is defeated appropriately to the form of struggle. The player usually gets a few chances to defeat the Villain, traditionally (in stories and in games) three. Again, if the Hero is unsuccessful, a new plot can be introduced.
19. Liquidation	K	The original object of lack or object of villainy is recovered. This can be a separate function to the victory, and may require other characters helping in the recovery.
Section V: Return		
20. Return	↓	The Hero leaves the location of the struggle, usually by the same means as the arrival in the guidance function.
21. Pursuit	Pr	The Hero is pursued by the Villain or his underlings. The underlings are defined by their attitudes to the Villain and Hero, which can be based on attitudes pre-authored by the designer, or arrived at through the social simulation system.
22. Rescue	Rs	The Hero saves himself, or is rescued from pursuit. After this function, a new plot move can begin with a new act of villainy. The rest of the functions of the current move can be intertwined with the new move, or they can wait until the new move is over, or they can be omitted.
23. Arrival	o	The arrival of the Hero either home or to a new kingdom, where the story will end.
24. Claims	L	The false claims are made by the character given the Falsehero role. This character can claim responsibility for the Hero's deeds, or claim the Hero did something he did. This introduces the use of deception, and the Falsehero is a character that the hero was previously friendly with.
25. Task	M	The Hero is assigned a difficult task. The task is like the struggle function, but the Hero is not directly opposed to a main character, and the task can be one of endurance or some feat of skill or strength. In computer games, tasks and struggles are abundant, but in more and more modern games, it is recognised that additional elements are required to express the progression of a story.

26. Solution	N	The Hero solves the difficult task.
27. Recognition	Q	After the task is solved, recognition of the Hero, exposure of the Falsehero, and/or transfiguration of the Hero is executed. Any combination of the three is possible. Recognition is achieved if the hero has been branded in the struggle with the Villain.
28. Exposure	Ex	The Falsehero is exposed when the Hero is recognised, or by his failure to complete a task.
29. Transfiguration	T	The transfiguration of the Hero can be done with a graphical change of the hero's appearance.
30. Punishment	U	The Falsehero or Villain, or both, are punished.
31. Wedding	W	The Hero receives his reward of a wife, a kingdom, untold wealth and power, a new hat, etc. The character with the Princess role will be the hero's wife, and for the OPIATE game engine, it was decided that a wedding function would be implemented by allowing the player to take control of the character in the Princess role. This rewards the player concurrently with rewarding the hero character, which is a technique that Tim Schafer uses in his games to enable player empathy with the Hero [Schafer 2004].

Figure 4.3: Propp's Character Functions

Each plot *move*, as referred to in section 2.4, is made up of some combination of these elements. Moves can be combined flexibly, and a number of the functions above refer to the introduction of new plots depending on the hero's reaction. These events will cause a new move to be introduced. Propp's 'Morphology of the Folktale' [Propp 1968] contains an appendix with 44 tales transcribed into series of functions. He transcribed others, but included this subset which he says exhaust the possible variations of functions, and therefore are an excellent source of raw material from which to draw our desired expert knowledge on plot structures. I converted these transcriptions into scripts as text files. These scripts take a simple syntax with each script line representing one character function. The SD parses these scripts and uses them as its case library for the case based planning AI, detailed later in section 4.6. The scripts are skeletal plot moves, to be fleshed out with their enactions by characters. One common misunderstanding and criticism of Propp's work comes from the assumption that he intends these functions as the most basic primitives of stories. Instead, Propp refers to the functions as *species*, made up of structures, each with any number of *genera*, of which he gives some examples. The scripts consist of some functions that are open to interpretation by the SD, and some that are tightly defined and need to be enacted according to the script. An example script is seen below in Figure 4.4.

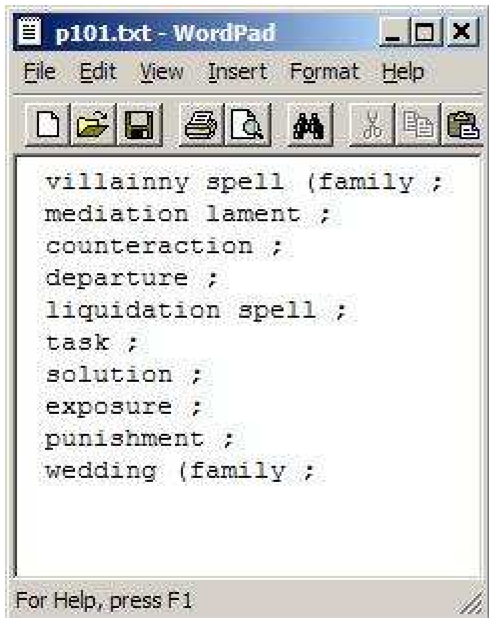


Figure 4.4: A Simple Plot Script

Each script line contains a maximum of three parts, called the **type**, the **verb**, and the **noun**. There are 31 types of character function, 26 possible verbs, and 9 possible nouns, as seen below in Figures 4.5 and 4.6. The verbs were designed by myself to be both descriptive of the many different genera of character functions described by Propp, while also fitting OPIATE's game worlds. Each verb can be carried out by using certain actionObjects. The nouns are used when a function requires another entity in the gameworld apart from the NPC that is enacting it, and the player character.

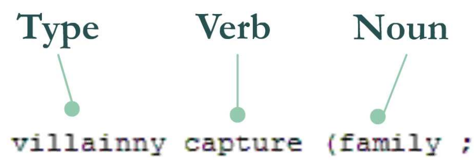


Figure 4.5: Script syntax

Verbs	Nouns
Attack, Expel, Transport, Quest, Spell, Capture, Fly, Help, Hide, Torment, Free, Lure, Transform, Engage, Kill, Appear, Track, Resuscitate, Marry, Test, Take, Lament, Show, Sold, Find, Pre-, Not.	Family, Wanted, Helper, Princess, Needed, Object, Captive, Hero, Action.

Figure 4.6: Script Parameters Adapted from Propp

Each of the 31 function types is associated with a role. There are 9 roles defined, 7 of which correspond to Propp's 'spheres of action' of the **Hero**, **Villain**, **Mediator**, **Donor**, **Helper**, **Princess**, and **Falsehero**, and another two, mentioned by Propp; the **Family** role which was included as part of the sphere of action of the Hero, and the **King** role which was included in the sphere of action of the Princess. Each role with its associated character functions is shown below in Figure 4.7. The roles can, with the exception of the Hero role, be assigned to any available character that satisfies the appropriate criteria. The roles are dynamic, with characters capable of being assigned a number of roles, and the roles can be occupied by different characters at different times.

Roles	Functions
Hero	3. Violation 10. Counteraction 13. Reaction 18. Victory 23. Arrival 26. Solution
Villain	4. Reconnaissance 6. Trickery 8. Villainy 16. Struggle 17. Branding 21. Pursuit
Donor	12. Test 14. Receipt
Helper	15. Guidance 19. Liquidation 20. Return 22. Rescue
Mediator	9. Mediation 11. Departure
Princess	31. Wedding
False Hero	24. Claims
Family	1. Absentation 2. Interdiction 5. Delivery 7. Complicity
King	25. Task 27. Recognition 28. Exposure 29. Transfiguration 30. Punishment

Figure 4.7: Role and Function Associations

These correspondences have been minimally adjusted from those described in Propp's work, to allow for the distinguishing between the Hero and Family roles, and the King and Princess roles. When an NPC is assigned to a role, they can then be given a storyGoal to carry out a function that belongs to that role. The function type is chosen based on the temporal stage the current move is in, and the verb and noun are either taken directly from the script, or extrapolated from the available resources in the game world. There is a constraint-satisfaction-like system that does this, yet is not as mathematically rigorous as the work generally done in that field. This system is implemented as a rule-based component more fully desribed in section 4.6.3 and Appendix B.

4.3.2. Character

Characters are modelled as semi-autonomous agents. This means they carry out some actions and deliberations without instruction from the higher-level SD agent. The characters are modelled on a number of different levels, shown below.

- Low level - for example, collision detection which avoids nearby objects and characters as they get too near. There is a state machine which keeps track of what the character is doing.
- Idle behaviours - When an NPC does not have a goal to achieve, they can execute behaviour such as patrolling around a house or following another NPC. These behaviours are assigned by the story author.
- Social simulation - the NPCs use a basic gossip algorithm and inform other characters of events that have happened to them, including other gossip events. The order they have met the other characters can initially dictate who will be the target of their gossiping, until Attitudes have been formed through the story events, or by the explicit authoring of Attitudes before the game starts. Section 4.3.3 and section 4.5 will more fully describe the social simulation.
- Attitudes - characters develop ratings for characters that interact with them, and characters that they hear about via the gossip algorithm. They remember the specific events that caused these rating changes, so they can gossip about them.

Attitudes are one-dimensional, (like/dislike), but the model could easily be extended into more dimensions, such as Crawford's extensive multi-dimensional model [Crawford web].

- Targeted behaviour - The SD agent assigns storyGoals to characters, and the characters search for the object of a goal and execute it. When targeted, a character will not execute idle behaviours.

The model is not advanced, and more complex systems are found in many games, but it serves our plot-focused needs, in line with Szilas' contention that characters in a story must primarily serve the plot [Szilas 2002]. Characters are all based on this model, including the player character. Although the player directs the choices of their character, social ratings and attitudes are formed without the player's control, according to the same rules as the NPCs. This is both to enable a player to switch between controlling different characters, and to help the SD react to player actions without having to model the emotional state of the player themselves, an approach being attempted by some [Szilas 2003]. The idle behaviours are disabled, but the player character can be assigned a goal through the player interface. The abilities that characters have are dictated by *actionObjects* that inhabit the world. ActionObjects can be picked up, dropped, and used on both other characters, and objects in the world. The author defines what objects correspond to each verb that will be required by the SD storyGoals, and also defines attributes of the actionObjects, such as how they will effect attitudes when used. ActionObjects need not only represent embodied entities in the world, and it was found to be useful to treat abstract actants in the game as actionObjects, for example, a problem that a character may have.

4.3.3. Thought

Aristotle's thought layer concerns everything that *is asserted to be or not to be* in the course of the action. There are events that take place in the game world, divided into storyEvents, and just plain events. StoryEvents are those that take place because the SD initiated a storyGoal which was carried out, and those that the player character causes in response to a storyEvent. Everything else, including gossiping, player actions not directly related to the plot, and NPC events that are generated through the forming of attitudes, are just events. However, the gossiping system does not distinguish between the two kinds, and the attitudes are a system whose dynamics are based on the story events being fed-back into the system to influence the character dynamics, which will in turn influence the SDs deliberations on what plot to next introduce, and how to enact it. In this way, an interactive complex system is created, which allows a player to determine the way that their character is perceived in the artificial society.

Figure 4.8 shows this feedback cycle in more detail.

1. The SD evaluates the most suitable character for a given role, and assigns the current storyGoal to that character.
2. That character then performs the action, causing an event that nearby characters are alerted of so they can act as witnesses.
3. This event causes a change in characters' attitudes about the two characters involved in the event.
4. The player can react to the event, which changes attitudes further.
5. Characters gossip about the event to characters that they have positive attitudes about.
6. The Story Director bases future deliberations on these attitudes.

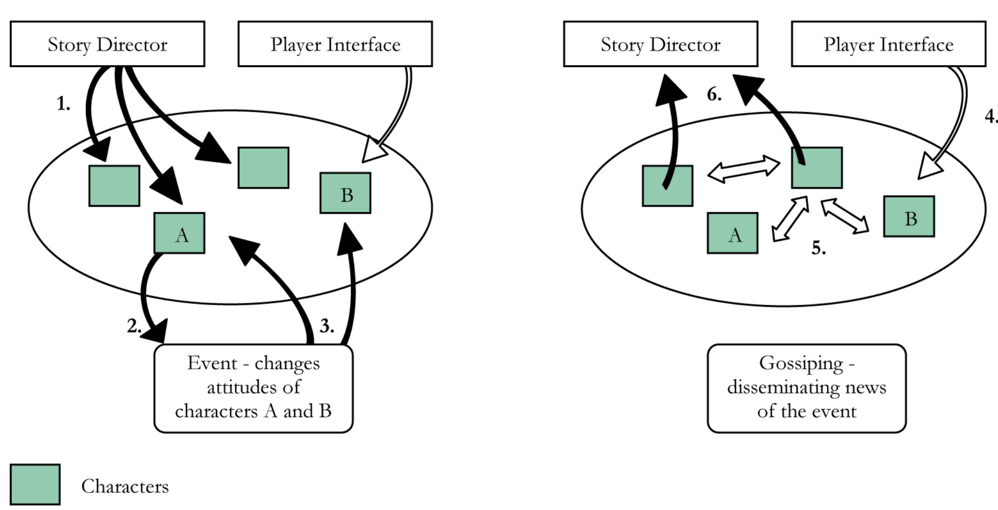


Figure 4.8: The Dynamics of Events & Gossiping

This architecture is fundamental to the unfolding of the story from the player's point of view, and means that the SD abstract plot functions can be based in an interactive social simulation, with causal links between events that otherwise are purely temporally ordered. This symbiosis of the two levels of character and plot is vital to the story as a whole.

4.3.4. Diction

In order to carry out the actions that are required of the characters, a crude dialogue generation system is used. This functions well enough to to refer to the objects, characters, and verbs in the game world. All of these are given a name, and the dialogue is generated based on simply stringing them together in single sentences. Although this approach is very limited, it could be extended, incorporating more expressive natural language generation. The player can only talk to other characters using references to objects and characters that they have encountered in the game, using the same sort of speech acts that the NPCs do.

Adventure games, both text, and graphical, have used dialogue to great effect to convey characters and situations, and automatic dialogue is not currently able to match this expressivity. Using flexible sentence fragments in a dialogue engine could help this, but since many games are now using pre-recorded speech for dialogue, there is a fundamental incompatibility with all current forms of istory and recorded speech. I don't see a general solution to this emerging in the near future, which allows the artistry of voice acting along with the flexibility of istory, but an imaginative use of speech fragments and patient voice actors could go some way towards it. In OPIATE, simple text fragments that are combined with references to game world entities is enough to get across the events of the story, along with simple animated displays of the use of actionObjects. Some pre-authored generic lines were also used, as recourse if the SD cannot find suitable verbs or nouns currently available in the game world. This was not planned initially, but was found to be necessary in some cases. For example, to execute the **pursuit** funtion if a suitable action is not available, a character can simply chase the hero character, saying 'You'll never escape!'.

4.3.5. Song

Sometimes known as the 'pattern' level, Aristotle used it to refer to the use of melody to give pleasure to the audience. Brenda Laurel [Laurel 1991] postulates that instead of seeing this melody as pertaining to purely aural phenomena, the use of pleasing patterns in the story is what Aristotle was referring to. The 'play within a play' in Hamlet is a well-known example of a sub-element that alludes to the larger plot. The use of repetition, of motifs and of plot devices, emphasises certain parts of a story and can give the audience the impression of a unified structure.

Fractals, the existence of self-similarity at different topological levels, can be found in many natural phenomena, and has limited applicability to artificial creations. Some poetry has been found to exhibit some self-similarity, and it can also be found in the work of artists such as Escher and Dali [Mandelbrot web]. To take advantage of this technique, it was proposed to include a method for making certain events take on structural features of higher-level plot functions. This could be done to introduce new characters, to emphasise events and to foreshadow. However, this was not implemented in the final system, as the higher-level plotting is highly dependant on the player's actions, and the SD plotting is done on a more immediate, short-term, basis. Thus, there would not be a simple way of including shorter elements than the plot moves that are already present. Reversing the direction of self-similarity would be an option for an implementation of this, however, and a higher-level plot could be planned based a combination of the current finer-grained plotting fragments. The higher level plot would then be dependant on the collection of plot moves that the player has experienced - and that they have caused - and would also have a fractal nature. This seems to be a compelling way of integrating Aristotelian patterns with OPIATE, yet was not implemented due to time constraints.

Since repetition is another way of patterning events, there are some functions that can be repeated, as a means of trying to convince the player to take action. The sequence of Donor functions - Test, Reaction, Receipt - is often repeated in Propp's corpus, as the hero often fails to achieve the proper result the first time around. Whole sections of moves can also be repeated, with tale no. 114 (frrom [Propp 1968]) repeating the donor and pursuit functions. This repetition is handled by the SD, which checks on the result of hero functions, and schedules repetitions accordingly, and repetition is also contained in the plot scripts themselves.

4.3.6. Spectacle

The graphical models in the games were all coded in native OpenGL procedure calls. Appendix C contains an example of the code used for models and animations. Because native OpenGL calls were used, animation and modelling is integrated. There are tradeoffs involved in choosing this approach over doing the modelling in separate 3D modelling software. It's easy to alter models and animations using the 'apply code changes' while the game is running, in Visual C++. This enables quicker implementation of the graphics, which were only a means to present the stories in a way that people could appreciate, and not a focus of this project. The models are not of a modern high quality, but this was not a design requirement.

Three demo games were developed to test the functionality of OPIATE. What follows is a description of the character models and map layout in the third and final demo game developed, 'Bonji's Adventures in Calabria'. The first two games will be briefly described later. The main character in the game is 'Bonji', the player character and hero of the story (Figure 4.9). Also pictured here is 'Snomm', a mysterious levitating witch. 'Trivmaj' is a magician who has control over the wind. These characters reside in and around an initial locale called 'Leptown'.

Snomm(left) and Bonji

Trivmaj and policemen

Figure 4.9: OPIATE Screenshots

After Bonji finds a way to escape Leptown, he finds himself in another strange locale with a Party happening, lots of balloons, and some strange mushrooms (à la Nintendo's Mario games). This area is inhabited by party-goers Mick, Nige, and Dec, along with Gilmer and the mysterious Spicter (Figure 4.10).

Mr Pertin and Mr Perft

Gilmer at the party

Spicter

Figure 4.10: Characters in the Demo Game

Somehow, Bonji finds a magic carpet, and enters the third locale, covered in grassland and including the giant, 'Gronk', and a tribe of Jawas (Figure 4.11) – as seen in

Star Wars by Lucasfilm. The main challenge in this final area involves removing a river dam that is preventing a large tree from getting enough water to survive.

Gronk | Cerebus © Dave Sim

Figure 4.11: OPIATE Screenshots

There are 28 characters in total, spread through 22 locations. The layout of all the locations in the game is shown in Figure 4.12 below, divided into three locales. Bonji will travel from the first locale to the second by means of a 'whirlwind' object, and from the second to the third locale by means of a 'magic carpet'. These objects are present in the game world, and can belong to any of the characters. The whirlwind initially belongs to Trivmaj, who is cast as the Villain in the first locale, although this can change. It is a matter for the SD agent along with the character dynamics to determine when and how these transference events happen. There is free movement within each of the three areas, and limited movement between areas.

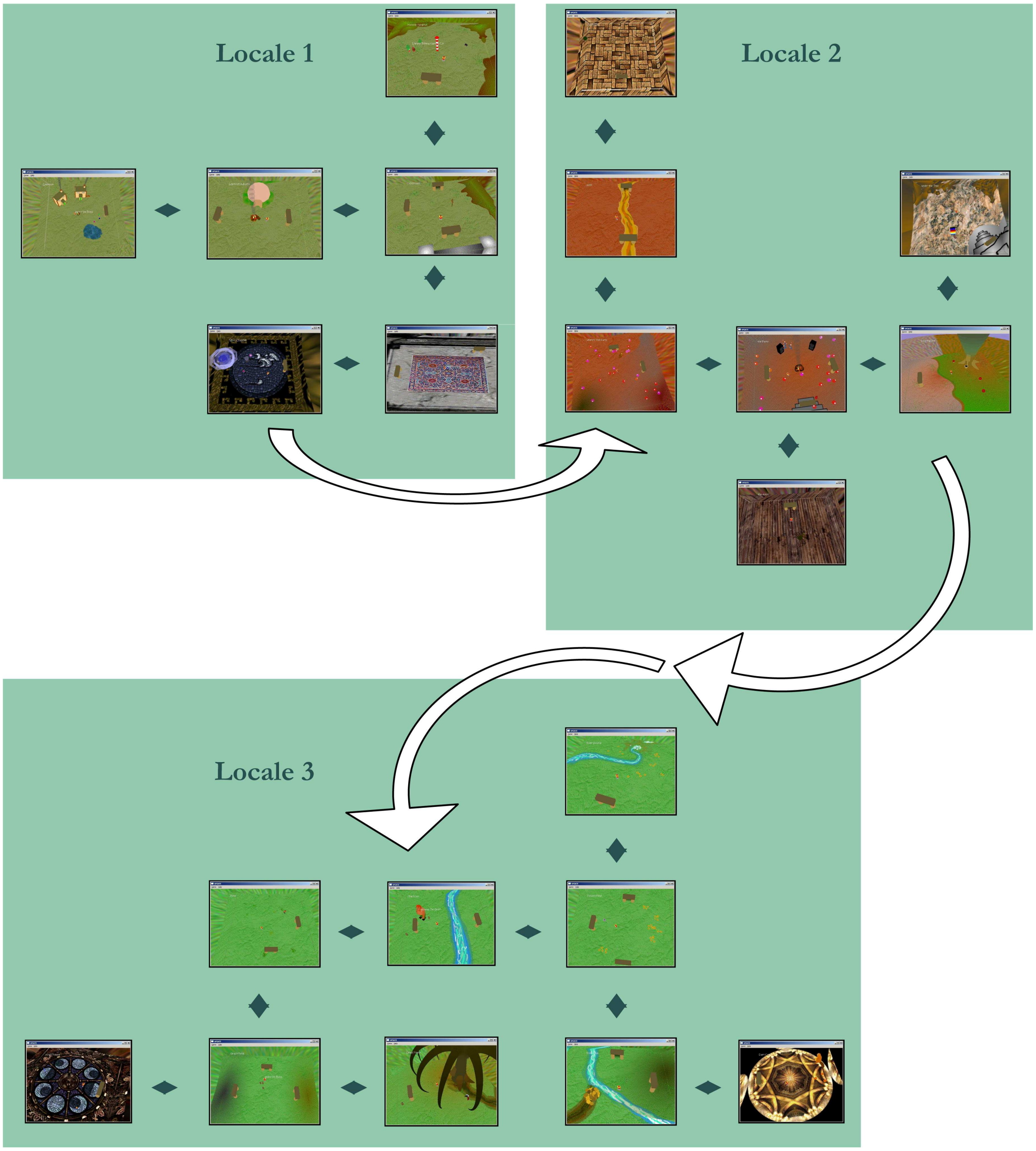


Figure 4.12: The Map Layout of 'Calabria'

Aristotle's meaning for spectacle was different to that used here, even taking account of the switch of context from theatrical drama to computer games. By spectacle, he meant the events that occur in a story that have a strong visual element to them, such as a romantic firework scene in a love story movie. However, after the presentation of these visual aspects of the design, the flow of an OPIATE story, from the point of view of a player, will be clearer to the reader. With the Story Director agent controlling the plot, the most effective way to author a story for this system is to define the characters, locations, and possible interactions, so as to influence the overall progression of the player, leaving the details of how the unfolding of the story occurs to the SD's storyGoals, and the game's dynamics.

The transference events between locales are strongly visual, and these could be considered Aristotelian spectacle, along with an area in the 'party' locale, where balloons are strewn around and can be kicked with some physically simulated bouncing and colliding(Figure 4.13). This was put in the game as a way of giving the player something to see, and the balloons can also be used as objects that characters want. The visual is the lowest Aristotelian level on which the story takes place, and serves the purposes of the higher levels, yet it has a considerable impact on the player's experience of the story. The visual motifs used in the game have the biggest influence on the player's conception of the storyworld, with each character seeming to have its own personality. Spectacle resides at the end of the formal chain of storytelling, as it is the final layer that is informed by all the others.

The magic carpet | Gilmer & the Balloons

Figure 4.13: The Use of Spectacle

4.4. Architecture

The overall architecture of the system takes account of the requirements and functional specification presented above. The Story Director(SD) handles the plotting, goals of characters, initiation of story events, and some patterning of events. The game engine handles the social and lower-level character dynamics, the gossip events, the dialogue generation, and the graphical display. The player interface controls the movement of the Hero character, and its interactions with characters and objects.

The SD uses a case-based planner, using as a case library Propp's corpus of 44 story scripts (seen in Appendix A), with a total of 80 plot moves therein. Story cases have either 1, 2, or 3 moves. Each move is represented as a case. The SD keeps track of the currently active move cases, the current function of each move, and periodically evaluates the game world state to update the assignation of Roles. The game engine updates the characters, handles events, and draws the current location of the hero character. The overall architecture is shown in Figure 4.14.

Although the OPIATE system was initially implemented as a single-player game, a multiplayer networked extension to the system was designed and implemented. This was because although the development of istory is attractive to allow player freedom of choice in single-player games, the paradox of this is that the player will still have a linear experience of the story, no matter how they choose to play through it. A more compelling motivation for the development of istory can be seen in the currently expanding MMOG (Massively Multiplayer Online Game) market. Players of these games log onto a shared gameworld and experience the game along with thousands of other players, in typically immense environments.

A big stumbling block that story authors for these games have to negotiate lies in the fact that players can each choose their own path through the story, with their characters teaming up and falling out with other player characters as they see fit. In order to tell a story in this environment, authors have resorted to a number of mechanisms, such as: (a) 'canned quests' that any player character can receive from an NPC, (b) large-scale plot arcs, that players are informed about through exposition, that cannot be interacted with significantly. (c) Devices that govern NPC's loyalties based on a player character's group affiliations. These mechanisms often lead to experiences that are not very story-like. A promising use of a system like OPIATE would be as a story director for the NPCs in a MMOG, which reacts dynamically to player characters' attitudes for each other, and for NPCs. The architecture of OPIATE seems to be conducive to this, so a networked version of the engine was developed. A client logs onto the server, which updates the game world and SD, and controls one of the characters in the game via a 'simulacrum' client-side world. This is updated through a connection manager which communicates via UDP sockets with the server. Figure 4.14 shows the overall system architecture, incorporating the multiplayer component. The single player game has the same architecture, except that the character interface is plugged directly into a server-side game character. Another key difference is that the SD has to be aware of a number of different player characters, and schedules a different set of story moves for each one.

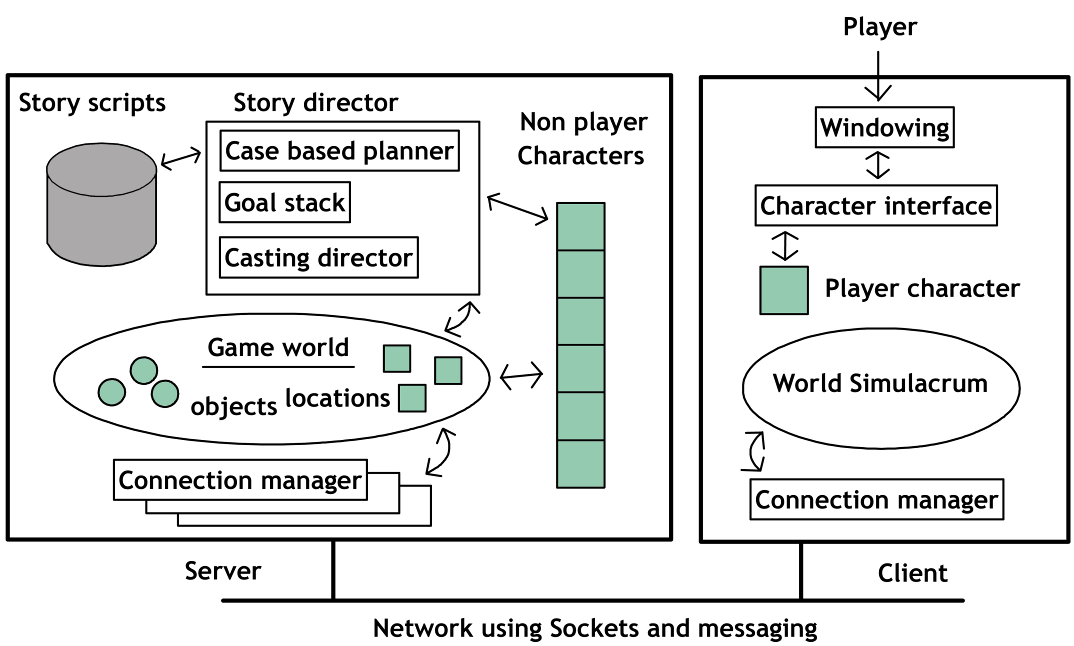


Figure 4.14: Overall System Architecture

4.5. Game Engine

The 'main loop' in computer games updates all entities in the world, draws them, and processes player input. The entities that need to be updated in OPIATE's game engine are the characters and the objects. Characters are the most important, and only some objects need to be updated, as most active objects are manipulated purely through characters. The character modelling, as shown in section 4.3.2, is divided into low-level, idle, gossip, attitudes, and goal-directed behaviours. The character architecture is shown below in Figure 4.15.

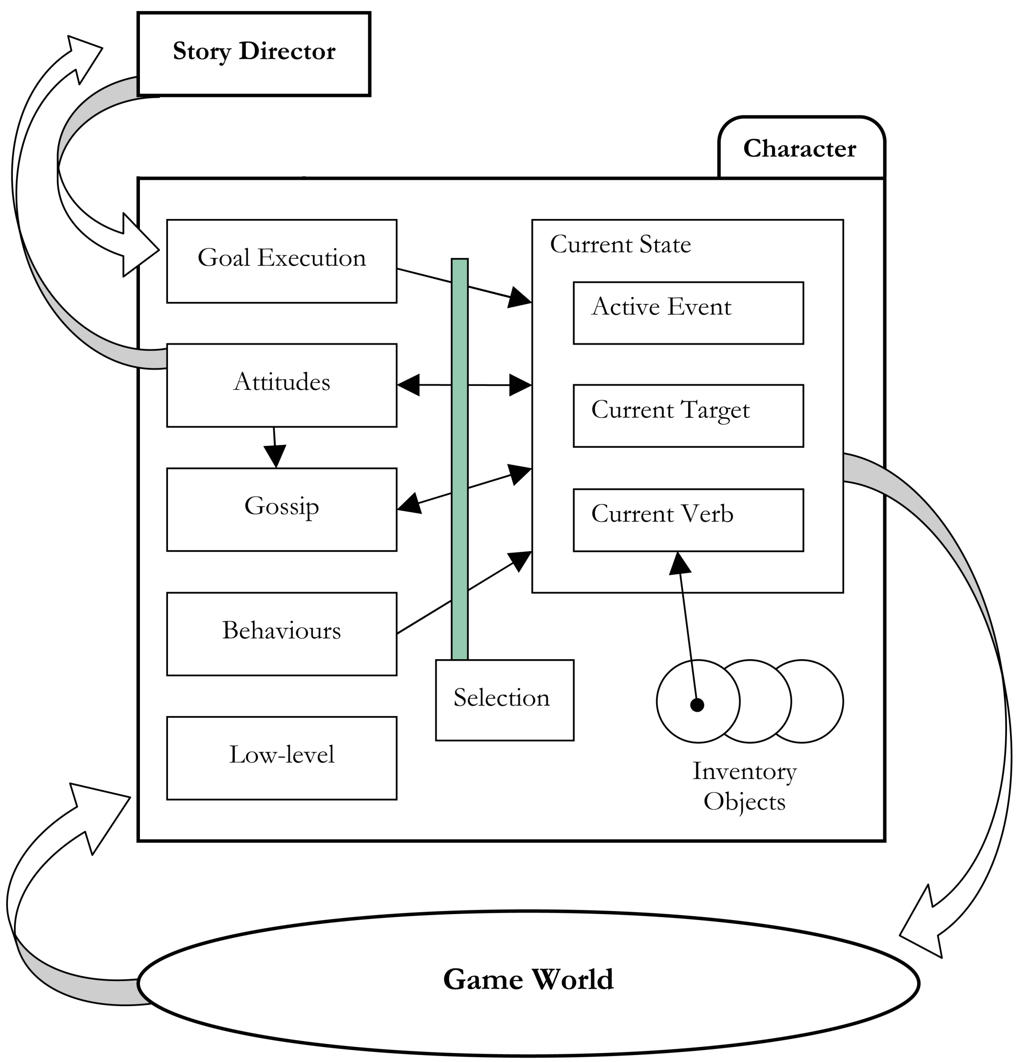


Figure 4.15: Character Architecture

The selection mechanism simply chooses the highest level behaviour that is currently pending, and executes it. If a goal is present, social and gossip events are disregarded, and if a social or gossip event is present, the idle behaviours are disregarded. Goals are pursued by simply navigating through locations until the target is found. The search is basically random. When a character is looking for something, it chooses a random portal from those available, and enters it, providing that it has not just arrived through that portal. This means that characters can seem to be randomly moving through locations, occasionally doubling-back on themselves, yet this is more efficient than maintaining a search algorithm. As there are no more than three portals in any location, and up to only nine locations in a locale, characters always find their targets with no difficulty. The doubling-back is not entirely a bad thing, as this effectively results in a type of breadth-first search, instead of a depth-first search, and can also make the characters seem less like deterministic machines.

Once a character can see its target, it will execute it based on the parameters of its goal. If the goal is to use an actionObject, then this will be done, along with a display of any generated dialogue from the use of that actionObject. The range of goals that can be executed are all initiated by the SD, and will be described in the next section. Because different character functions can be composed of the same actions, there are less goal types than the 31 character functions.

Attitudes encompass a character's memory of all previously witnessed events, categorised into events concerning each of the characters that they have met or heard about. Each attitude consists of a reference to the subject character, an integer value between -10 and +10 (representing dislike and like, respectively), along with a list of all the events that have contributed to that value. These attitudes determine the gossiping, and if a more extreme attitude (set to either -5 for dislike or +5 for like) is held, the character can spontaneously perform either positive or negative actions on the subject character.

To illustrate using hypothetical characters A, B, and C, Character A performs gossiping by taking a recently witnessed or heard-about event, and talking about the subject character of that event, character B, with the target of gossiping being a character C for whom character A has a positive attitude. The gossiping act results in the generation of dialogue, which relates the event to character C. Character C then modifies its attitudes to character B, depending on whether they like character A. Equation 4.1 below shows how this works, with XaY denoting character X's attitude for character Y.

$$\text{given: } AaB,\ AaC > 0,\ CaA,\ CaB_0$$
$$CaB_1 = CaB_0 + \left(AaB \times \frac{CaA}{|CaA|} \times W_A \right)$$

Equation 4.1

So, if C likes A, it will add A's attitude to its own, but if not, it will add the opposite attitude. The term $W_A = 1$ if the event that is being gossiped about was witnessed by, or if the event directly concerned, character A, and $W_A = 0.5$ if character A was informed about it via another gossip event. These simple rules result in quite a dynamic system, with 'cliques' forming, and news of events being disseminated through the set of characters. Whether the model is completely accurate in simulating human social behaviour is beside the point, as such a model could be extended almost indefinitely, and for our purposes it is enough, in that it provides a simulation that can be interpreted by the player as a dynamic social system.

The author of the storyworld can define some 'back-story' events that initially dictate attitudes, or characters can start with no events and attitudes. These attitudes are functionally exactly the same as those generated by in-game events, and can be overridden as the dynamics of the social simulation system progresses. Another means of authoring characters is the use of idle behaviours. Characters can take a number of types of these behaviours, from guarding a location, to following another character around, and they can also be reactive behaviours that only occur if certain conditions have been met. It was also found to be useful to allow new behaviours to by conferred through the use of certain objects. This can happen if, for example, character A uses some object to capture character B. A behaviour to follow character A will then be added to character B. Some behaviours can also be higher priority than others, overriding gossip goals.

ActionObjects are the authored means of interaction, and can be picked up from locations, dropped in locations, given to other characters, and used on characters and objects. Each actionObject has its own animation, and effects attitudes in its own way. Some of the actionObjects authored for 'Bonji's Adventures in Calabria' are shown below, in Figure 4.16, along with the integer value that they change a character's attitude by. One notable exception to these actionObjects is the 'problem' object, which can be defined by an author as a requirement of a character for a certain other object. These 'problem' objects can be created, if necessary, by the SD agent, and this is the only way the SD can directly influence the storyworld apart from the assignation of storyGoals, if a plot move requires it, and there are no author-defined problems.

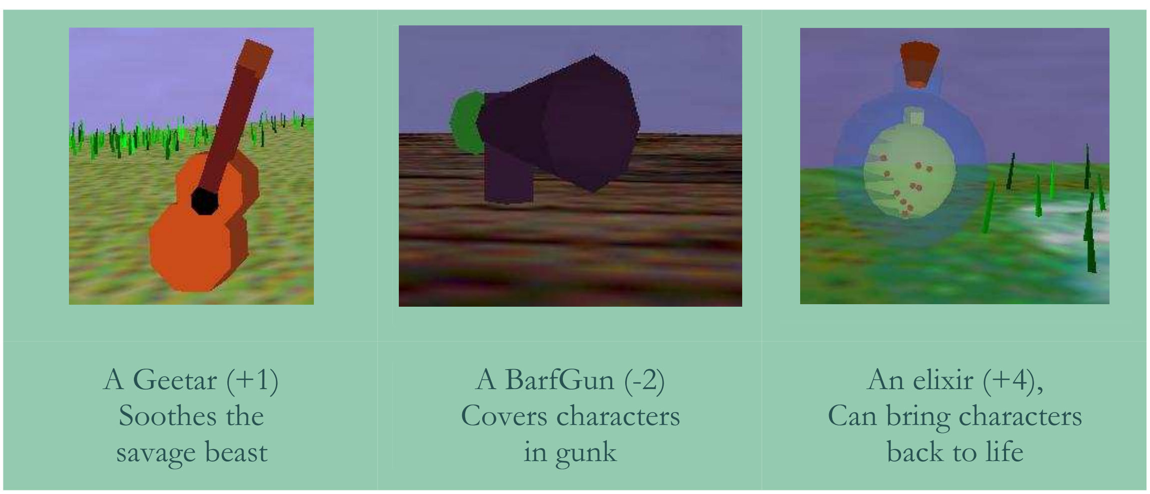


Figure 4.16: ActionObjects

Low-level activities primarily concern movement around the locations, steering to targets, and collision avoidance, along with perception. Collision avoidance uses an algorithm like those used in A-life flocking algorithms. The mechanism steers away from characters or objects that are too close, with more accentuated turning depending on how close the entity is. These mechanisms are shared by the player character. The steering mechanisms implemented in this work correspond closely to the *pursuit*, *wander*, and *collision avoidance* models that Reynolds describes in [Reynolds 1999].

The interface allows the player to directly control the Hero's movement using the keyboard, and whatever the Hero is currently pointing at is the 'focus' object. This object can also be selected on screen with the mouse. The focus object is the target of any commands that the player gives. Commands are given through the character control window, shown in Figure 4.17. The game engine displays text over characters' heads to show speech acts, which are generated using simple sets of rules. In the case of a story event, one that is initiated by the SD, the text is overlaid on the screen to emphasise the dialogue. This can be seen in Figure 4.18, below.

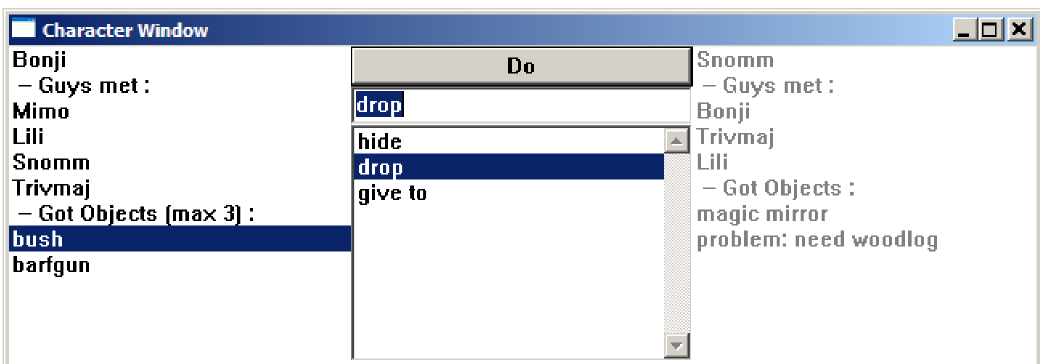


Figure 4.17: The Character Control Window

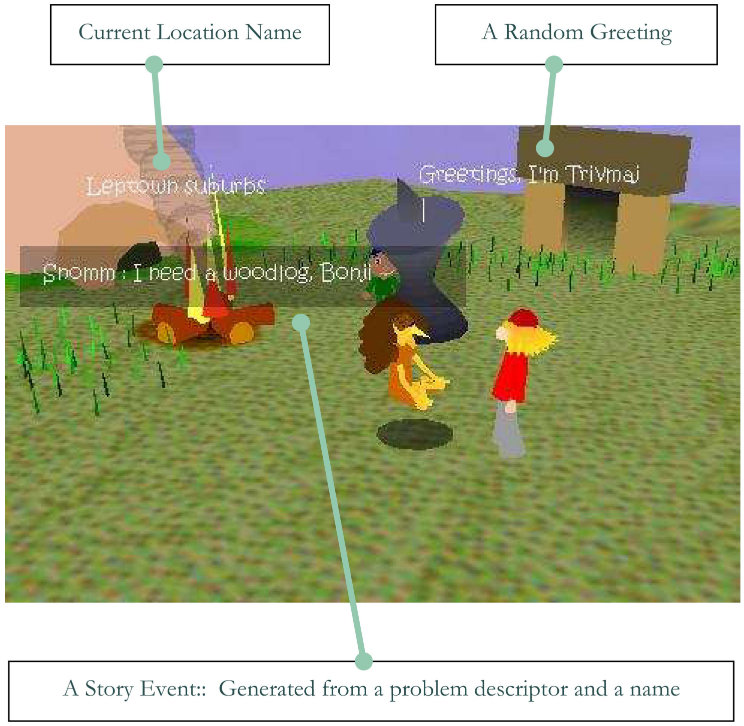


Figure 4.18: Dialogue

4.6. Story Director

The SD agent is the main focus of this work, as it is responsible for all planning and scheduling of storyEvents. Any innovation in the OPIATE system is to be found here, although the design of the system relies heavily on drawing concepts from other projects. Although Propp's work, and its correlation with the Oz project, was my initial inspiration for the project, the concept of using case-based planning to achieve the synthesis of plot-based and character-based istory is vital. Case based reasoning is less well known than ANNs and GAs in the general perception of AI techniques, yet it has been applied to storytelling in 1992 by Turner, although this was for a non-interactive system. There is more current interest in using this approach for interactive systems, yet it is relatively recent [Gervás et al 2004]. The overall vision of the OPIATE system as a storytelling medium was governed by the mechanism of using CBP (I) with Propp's plot scheme (II) in an architecture similar to the Oz architecture (III), within a social simulation that is interactive (IV). This combination of these four paradigms is what I propose the relevance of this work consists of.

The design of the SD consisted of using case-based planning mechanisms, alongside an implementation of Propp's rules and conventions as an expert system. In this discussion, we will first discuss the case-based planning system. The case library consists of 44 text files that were created by adapting Propp's corpus to the form of a script that is parsed by the SD when it is initialised at run-time. The process of this adaptation, which elucidates the case representation, is described in Appendix A. When the SD parses the scripts, each move of each script is converted to a distinct case for the CBP system. The links between a script's moves are preserved in the moves' representation as cases.

4.6.1. Case-Based Retrieval

The case based planning system uses a k-nearest neighbour algorithm to find suitable cases based on the heuristic shown below (Equation 4.2). The heuristic can be termed *suitability* metric, instead of the normal similarity metric used in case based systems. It finds the most suitable sub-plot to be enacted given the current state of the characters and storyworld, taking into account attitudes of the characters for each other, and for the player character. The core features that are used in this metric concern *roles* and *actions*. Roles are occupied by characters when they are enacting story functions, and the relevance of a character to a certain role is calculated based on past and present attitudes and memories concerning the player/hero character, and the other characters. Actions are enabled by *actionObjects* that occupy the storyworld, and allow characters to perform distinct types of interactions. They can all be picked up, given to other characters, and gossiped about.

$$Sn = \left(\sum_{i=1}^{Ln} (Wr * Sri) + (Wa * Sai) \right) / Ln$$

Equation 4.2

Where Sn is the suitability of case n, Ln is the length in functions of case n, Wr and Wa are the relative weights attached to roles and actions, and Sri and Sai are the suitabilities of the roles and action(s) present in function i. Sri is given by Equation 4.3, and Sai is given by Equation 4.4.

$$Sri = \sum_{j=1}^{\#C} Raj$$

Equation 4.3

Where #C is the number of characters currently available to the SD, and Raj is the relevance of character j to the role given by function i. The relevance of a character to a role is calculated based on a per-role basis, using a set of rules that have been created based on Propp's description of the roles. These rules are described later in this section. Some functions require two roles to be filled.

$$Sai = \sum_{k=1}^{\#A} Rak$$

Equation 4.4

Where #A is the number of actions currently accessible to the available characters, and Rak is the relevance of the actions to the actions required by function i. The action relevance values are binary, as an action object either fulfils the action given in function i, or it doesn't. The author of the storyworld must annotate each actionObject with the verbs that it can be used to perform, and the nouns that describe it.

4.6.2. Case Combination

Once a list of suitable cases are found, ordered on suitability using the quicksort algorithm, a decision is made to use the most suitable case, or combine cases to get a new one. If a hardcoded threshold suitability is reached, the former choice will occur, but if a combination of cases gets a better suitability, the latter will occur. Combination of cases is done on a per-function basis. As each function has its own suitability rating, the most suitable can be interchanged with less suitable functions in the target case. This is done by taking the longest suitable case and replacing its less suitable functions with equivalent, but better scoring ones from the second or third ranking cases.

An important element in combining cases is to maintain integrity of the structures when they are transferred, so Propp's *groupings* of functions are used to facilitate this. If a function is selected for transfer, and it has associated functions from the source case, these are also transferred to the target case. This can entail replacement of target case functions, so when the new case is constructed, it is reassessed for suitability. The groups are only of two or three functions, so this is not a difficult operation.

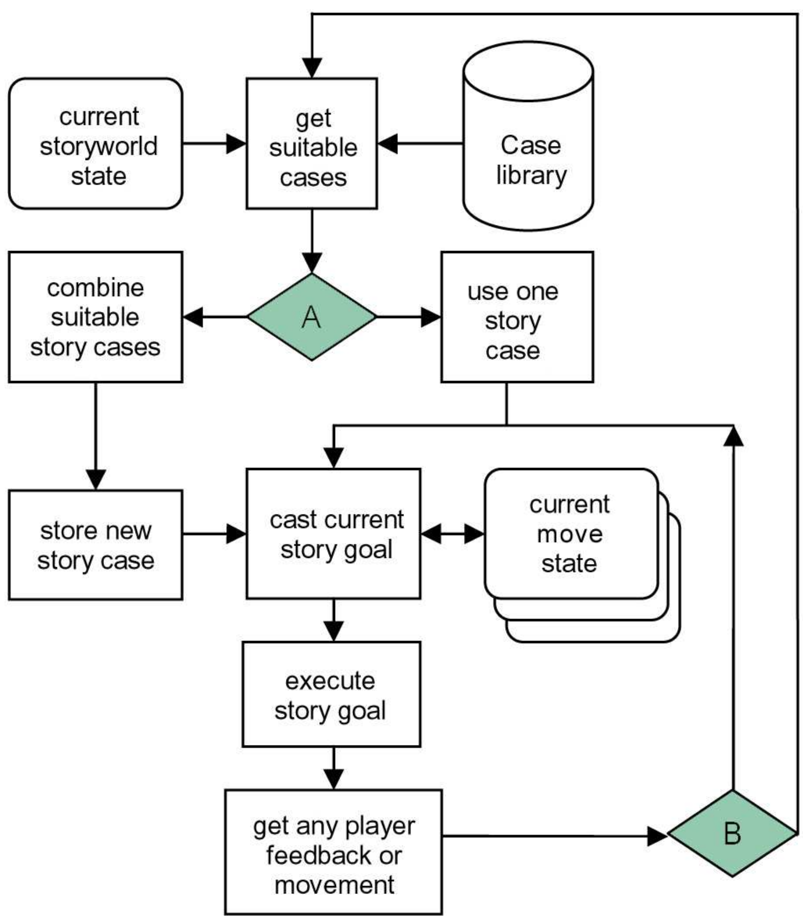


Figure 4.19: A Flowchart showing the Planning Process

At decision diamond A in Figure 4.19, a decision is made between using a single story move, or combining two or more moves to create a new case. This can happen if one move has certain functions that are suitable, with some that are not. Another move can supply more suitable functions to replace them. An example can be seen below, with the two moves in Figure 4.20. Both of these moves have elements that are suitable for the hypothetical storyworld state shown in Figure 4.21.

```
villainny expel ;
mediation transport ;
departure ;
donor ;
reaction ;
provision (helper ;
donor test ;
reaction ;
provision (wanted ;
return ;
```

Move (a)

```
villainny capture (helper ;
donor quest ;
reaction ;
provision ;
task ;
solution ;
wedding marry ;
```

Move (b)

Figure 4.20: Two Move Scripts

Move (a) features a Villain, a Mediator, a Donor, and a Helper, and the verbs 'expel', 'transport', and 'test', and a 'wanted' object. Move (b) features a Villain, a Donor, King, a Helper, and a Princess, and the verbs 'capture' and 'marry', and a quest object. These roles can be taken by the characters shown below, and the objects shown below can facilitate the character functions. Not all required verbs are present for either move.

The SD will choose to use the available resources (shown in Figure 4.21) to create a new move case, by selecting the two moves' most suitable functions, and combining them by replacing the most suitable move's least suitable functions. Because the verb 'expel' is not available for move (a), the villainy function from move (b), which uses the available 'capture' verb, will be substituted. Character B will be cast in the Villain role, and will receive the subgoal of acquiring object X, in order to carry out the villainy function.

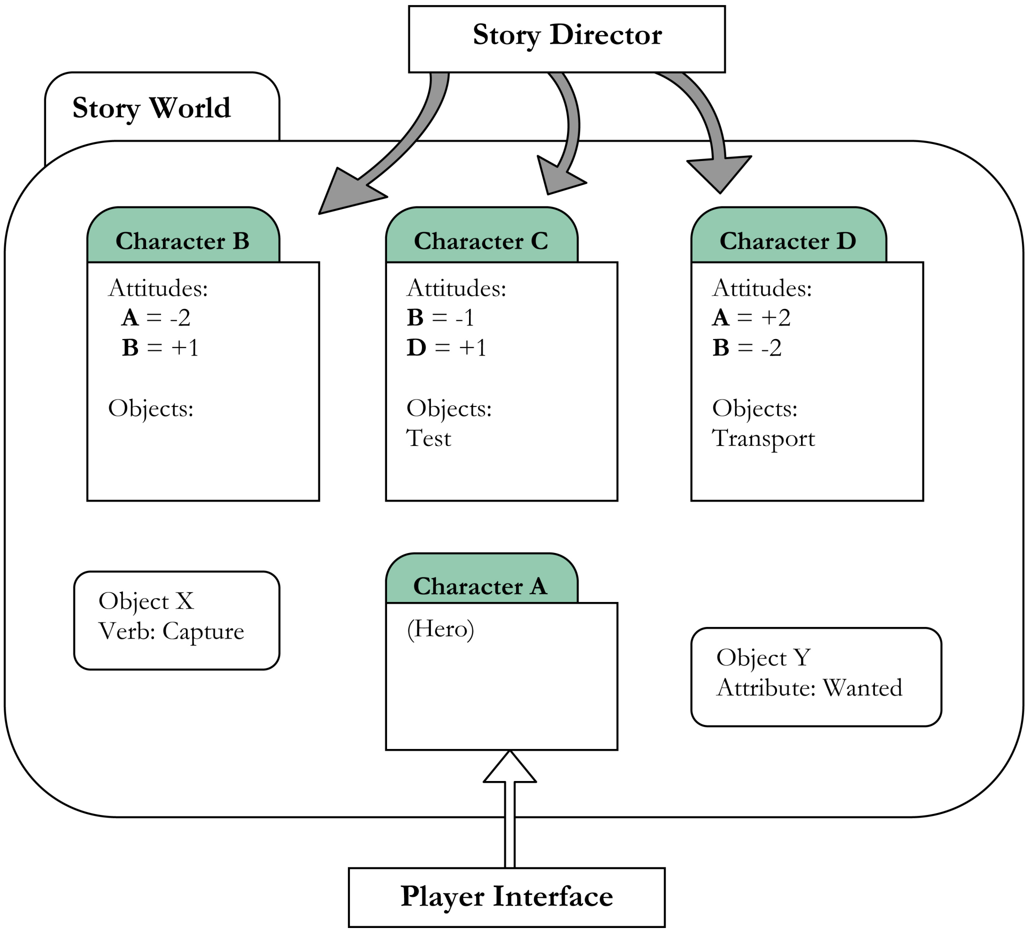


Figure 4.21: A Hypothetical Local Storyworld State

Propp's groupings of functions were not used to combine these cases, and only a single function was transferred to the new case. The *villainy capture (helper ;* line from Move (b) would be linked to the liquidation function if it was present, or a function that uses a Helper character. Groupings are done through these two mechanisms; the groups that Propp defined, and groups that share nouns. Groups that share nouns are grouped so that entities that were involved in previous story functions will be relevant for later ones. The groups that Propp defined are shown in Figure 4.22 below. If a function from one of these groups is selected for transfer into the new case, the others are automatically transferred, replacing any identical function types in the source case, to avoid repetition. A more involved example scenario is shown in Appendix E, which shows how these groupings are used to combine cases in OPIATE.

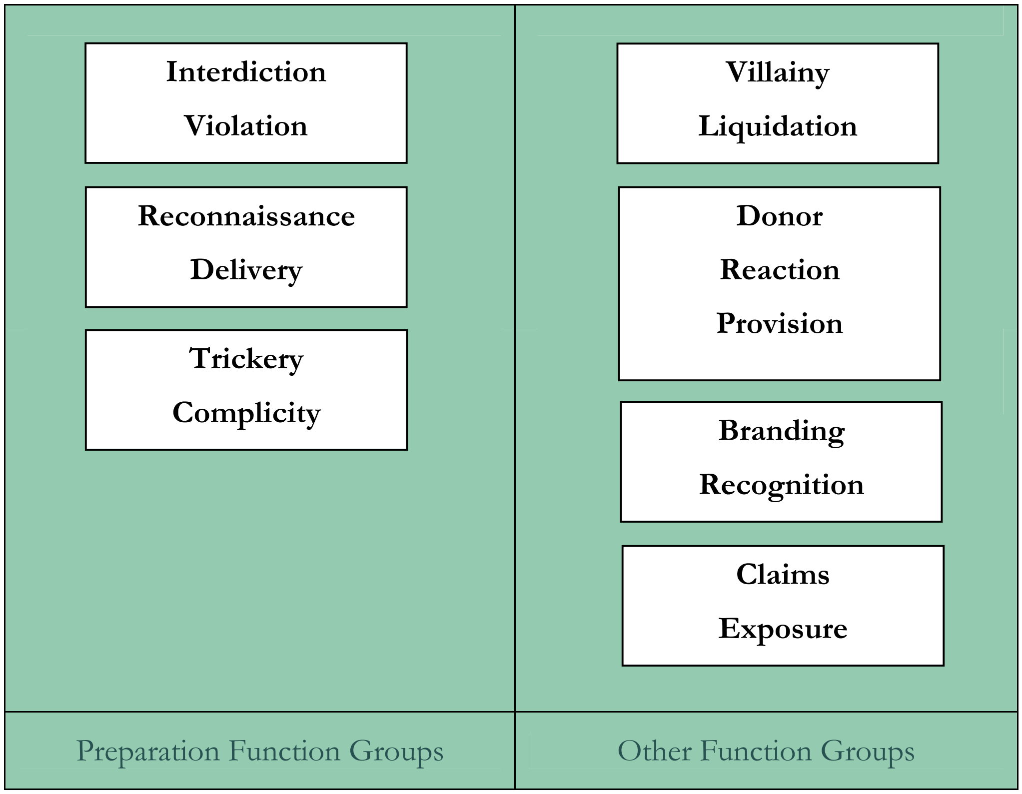


Figure 4.22: Character function groupings

4.6.3. Casting as Case Adaptation

Once a suitable case is selected, it must be converted from a list of abstract story functions into a series of events in the gameworld, interpretable by the player as a storyline. To this end, the story director uses a casting system which dynamically casts the NPCs into eight of the nine possible roles. The roles, previously defined in section 4.3.1, are: Hero, Villain, Mediator, Donor, Helper, False Hero, Princess, Family, and King. The hero character is always occupied by the player, even if they don't act particularly heroic. The usefulness of Propp's schema would be reduced if this was not the case.

These roles are cast as a move needs them. Once a case has been selected for use, it is cast as a move, whose progress is tracked by the SD. The roles required of the current move are dynamically cast as the move is being enacted, so that, for example, an NPC can take the role of a Donor, and later can be the Falsehero if the player/hero character falls out of favour with that NPC.

Casting is done using a set of criteria for each role. The villain role is filled by the NPC that opposes the hero most, with the most antagonistic attitudes between them, or else is an NPC close to that NPC. In this way, acts of villainy can be carried out by 'henchmen', depending on availability of NPCs. A Mediator can be any NPC that is available and nearby, even if the NPC is antagonistic to the hero. The Donor role can be filled by an available NPC that has not met the hero or has a slightly positive attitude. The Helper is filled by an NPC as the fulfillment of a positive previous encounter with the Hero. The Falsehero must be a NPC with a previous positive attitude to the Hero, who has either developed a negative attitude, or else has developed a positive attitude to the Villain. The Princess role is one that an NPC with a positive attitude to the Hero can occupy, or an NPC that has not met the Hero, but has been pre-authored as a possible Princess. The NPCs with positive attitudes to the Hero can all take the Family role, and the King is taken by a powerful character, that a large number of characters have positive attitudes to.

The specificity of these rules was formulated using a familiarity with Propp's work and its applicability to the game engine that has been developed, yet they could be editable through a toolkit if this system was to be used for other games. The rules are not arbitrary, and have been designed to maximise a sense of believability of the NPCs in their enactment of moves. These rules are the same as those used by the case based planner in determining how suitable character functions are, for use in the current game situation. A code listing showing these rules in more detail, in C++, is available in Appendix B.

Once an NPC is selected for a given function, the means of carrying it out is selected through a search of all actions available to the NPC. An NPC can be given a sub-goal to find and pick up a required object, or it can be given to it by another NPC. This is tracked by the SD, and when the sub-goal is accomplished, the storyGoal is assigned to the relevant NPC. The enactment of the story function consists of finding the target of the function, animation of the actionObject, and the generation of suitable text for dialogue. Descriptive or emotional text is not used, and syntax is kept extremely simple. Despite this, a story can be seen to emerge based on the simple dialogue.

4.6.4. Player Feedback

In the diagram showing the overall planning process, Figure 4.19 found in section 4.6.2, decision diamond B shows where player feedback is evaluated. A key feature of the SD agent is that it is focused on the player character, and must adapt its plans to the interactions of that Hero character. There are three distinct ways that this is done:

- **Move Selection** - Being aware of the NPCs' attitudes to the Hero and each other, and selecting moves that fit that set of attitudes.
- **Move Casting** - Using these attitudes to cast the NPCs in roles that reflect the way the Hero has interacted with them.
- **Move Initiation -** Initiating a new move selection if the Hero moves outside the scope of the current move. This is also done if the player consistently refuses to provide a positive reaction to one of the Hero functions in the current move.

The Hero functions are: Violation, Counteraction, Reaction, Victory, Arrival, and Solution. Repetition is a device much used in folktales, and Propp notes that the 'tripling' of hero functions is common. This device has also been used in computer games, with 'three lives' being a common feature of early games. Thus, the hero functions have three chances of being positive, with the appropriate functions being tripled to get feedback. The groupings of functions are shown below in Figure 4.23. If a timeout is reached, the function from column I is repeated, until the appropriate hero function from column II has been achieved by the player character. If, after three repetitions, the hero does not react, a new move is selected for enactment. The old move becomes dormant until the player provides the appropriate feedback.

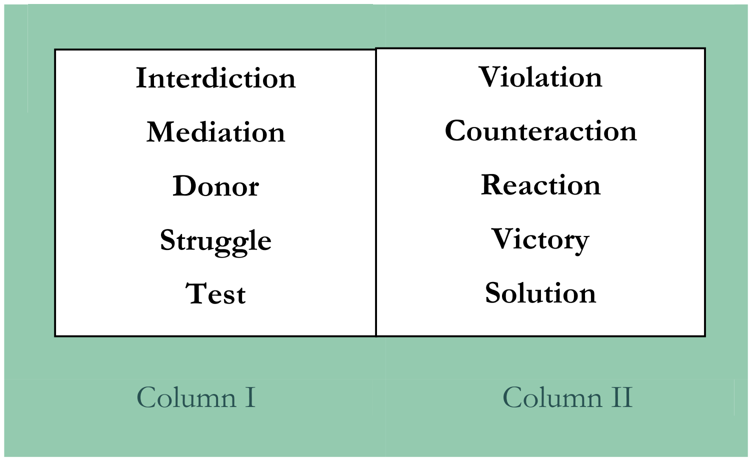


Figure 4.23: Player Feedback Function Groupings

4.7. Multiplayer

The decision was made to implement a multiplayer version of the game, as the application of istory could arguably be more useful to authors of massively multiplayer games. In these games it is impossible to create personalised stories for players using completely pre-authored scripts, therefore a multiplayer OPIATE was designed to test its applicability.

An important component of the multiplayer version is a copy of the game world that is updated on the client as required. At first, a version was implemented that simply sent interface commands to the server, which updated all game world entities, including the player characters. This was rejected because responsiveness of the game was negatively affected. A connection manager was used on both client and server sides to maintain consistency with the client-side and server-side worlds. Only the current location of the player is updated on that player's client.

The SD maintains an independent set of moves and related roles for each player, which are planned and cast just as in the single-player version. Because the same set of character dynamics and social simulation governs the SD's deliberations for each player, although each player's plot is different, they intertwine in their enactment. Each player controls a Hero character in its own story, but if Hero A has an enemy character C that is Hero B's friend, C will act in both Hero stories consistently with the player interactions. This is a powerful mechanism, and indicates that OPIATE has more potential in multiplayer environments, but the multiplayer architecture is not scaleable. As each new player requires its own moves and roles updated on the server, a server will not be able to handle thousands of players.

An improvement to this would be to have client-side story directors, which send messages to a central server to assign storyGoals to NPCs. This would be more scaleable, and seems like the best way to go; however, this was not implemented, as there were other problems with efficiency and network lag that were implementation-dependent. Another problem was maintaining consistency between the server and client game worlds. This is a common problem in online games, and a number of advanced solutions are used. A method of interpolation for determining character positions between the positions that are messaged was used, but resulted in some jerkiness of character movement. These problems indicate that in the future, if OPIATE is to be tested in a multiplayer environment, a separate commercial game engine should be used, which would eliminate many of these problems.

5. Implementation & Testing

5.1. Introduction

The OPIATE system was fully implemented in C++, using the Microsoft Visual Studio environment. This is a standard development environment, and the use of C++ facilitates efficiency in the often resource-hungry programs that make up computer games. Over 12000 lines of C++ code were written for OPIATE. The only exceptions to the use of C++ were the abstract plot scripts, that were authored as text files, based on my analysis of Propp's corpus which was taken from appendix III of [Propp 1968]. Appendix A shows how Propp's schematic representations were adapted to form these scripts.

Taking an object-oriented approach, the components of the architecture were divided into two main areas, the game engine, which incorporates a player interface, and the story director. Implementation proceeded by initially creating the game engine with a simple interface, with a skeleton story director implementation, and developing the complexity of the game world in tandem with the story director's capabilities. There were three milestone demonstration games developed and tested, each of which incorporate different features of the system.

This section will first describe the object classes that make up the system, and how they work together. Then the incremental testing will be described using the three demo games as examples, and finally, the integration of the system will be discussed.

5.2. Object Classes

5.2.1. Game Engine Classes

The game engine consists of game world classes and interface classes. Apart from the basic windowing and input handling, there are other interface classes such as the 'InfoPrinter' class, which handles the output of text from the world to the interface. The multiplayer component uses both the 'ClientManager' class, which manages the client list on the server side, and the 'WorldConnection' class, which handles the transfer of information about the game world on both the client and server side.

The primary game world classes are the 'World', 'WorldObject', and 'Location' classes. The 'WorldObject' class is inherited by the 'Character' and 'Object' classes, and incorporates information about the location, orientation, and size, of entities that inhabit the world. The 'ActionObject' class inherits from the 'Object' class, and represents objects that can be picked up and used by the characters. The game world class hierarchy is shown in Figure 5.1.

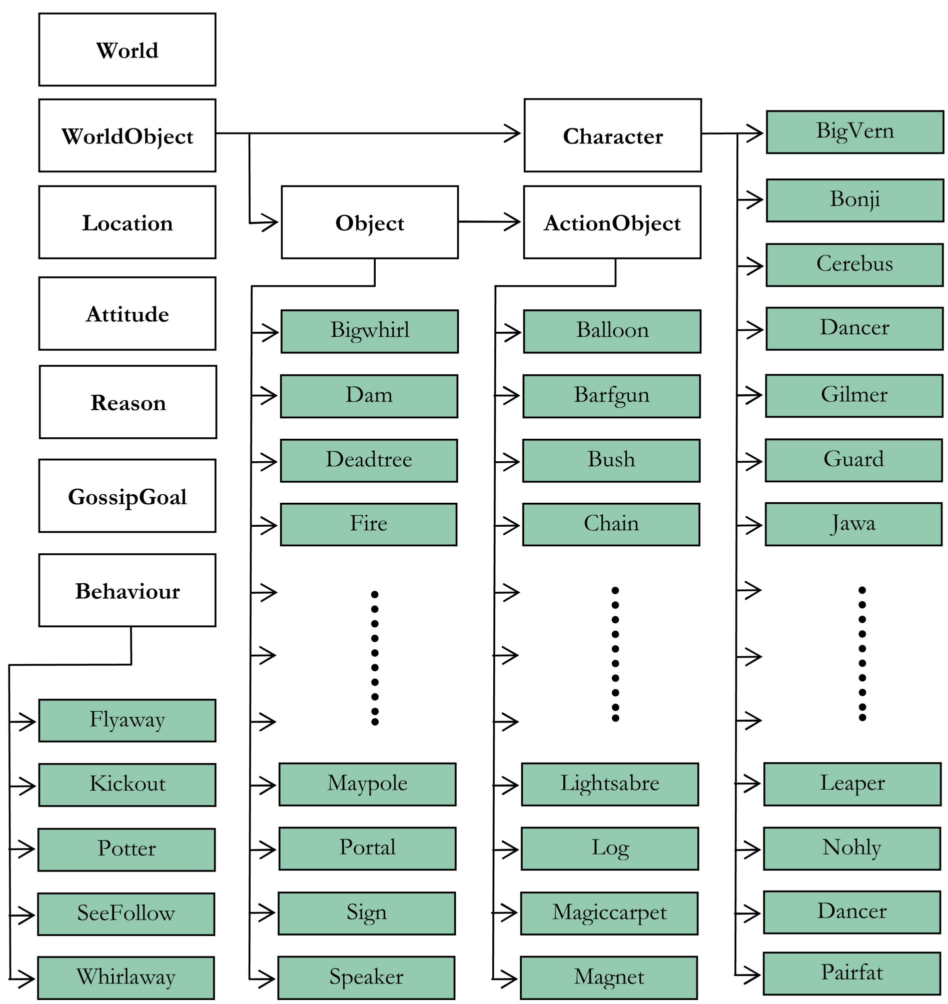


Figure 5.1: Game World Class Hierarchy

All game world 'WorldObject' and 'Location' classes have an *update()* and a *draw()* method, which are called by the 'World' class. The other game world classes, 'Attitude', 'Reason', 'GossipGoal', and 'Behaviour', are used by the 'Character' class in the social simulation algorithms. As described in the Design section of this thesis, the character simulation works on a number of different levels. Characters maintain linked lists of both Attitudes and Behaviours, and if a GossipGoal is active, it will consist of talking to another character about an Attitude, along with any associated Reasons. Reasons are references to game events that have been experienced, witnessed, or heard about. Behaviours are assigned to a character, and if idle, it will execute all its assigned behaviours.

ActionObject classes all have distinct *use()* methods, that dictate how the use of that object will effect the target entity. Some actionObjects can only be used on characters, some on objects, and some on both. The graphical display of all characters, objects, and locations, are done using native OpenGL procedure calls. Development of the graphical models was done using the 'Apply Code Changes' tool in VC++, which allowed models to be created and animated quickly, with alterations being made without having to stop the program and recompile the codebase.

5.2.2. Story Director Classes

The SD classes occupy three conceptual categories; classes that serve to give the SD a *world view*, and enable it to sense the dynamic game world; classes that serve the case-based planning mechanism; and classes that enable a plan to be adapted to the current game world, and enacted in it. These three areas of operation are controlled by the central 'StoryDirector' class, which utilises the three categories of classes to further its aim of directing a player-centric story. Below, Figure 5.2 shows the process relationships of these classes. Unlike in the previous diagram, the arrows do not represent class inheritance relationships, but process relationships that show how information is transferred through the classes as the system executes.

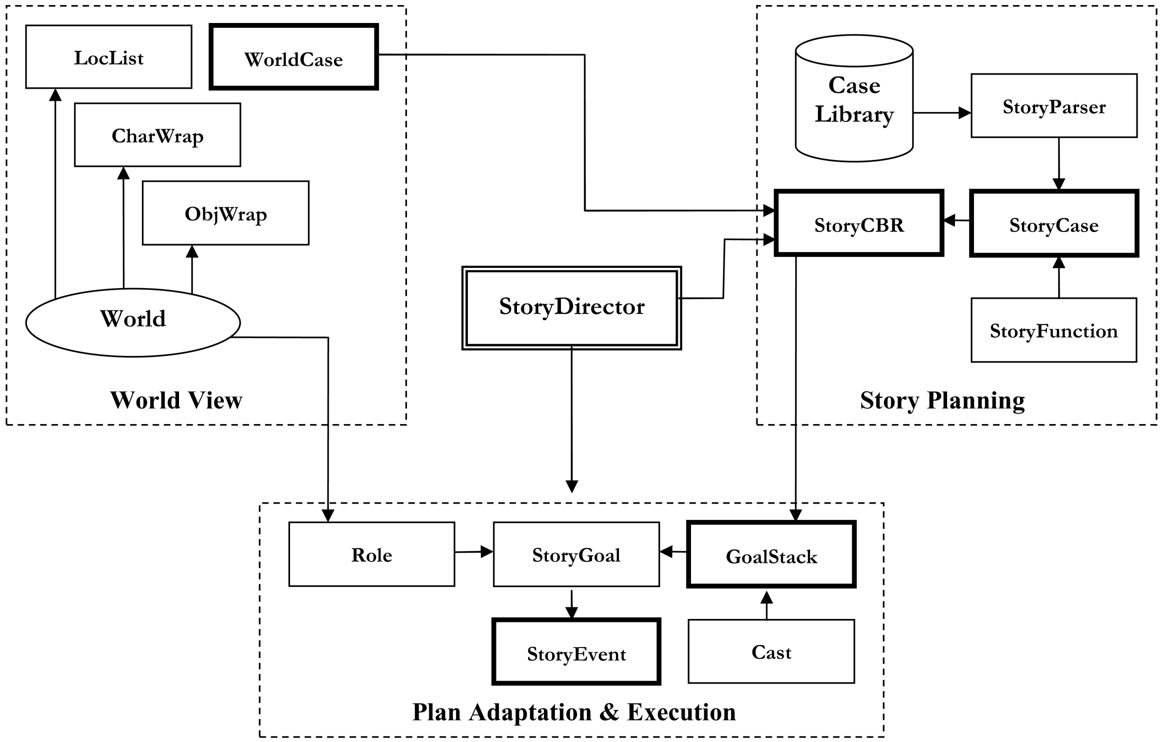


Figure 5.2: Story Director Classes and Process Relationships

The *world view* classes handle perception of the dynamic system of the game world, by encapsulating currently relevant information in wrapper classes. The current location of the player character is noted by a 'LocList', which maintains the currently neighbouring locations of the player character's location. The 'CharWrap' and 'ObjWrap' classes are linked lists of the available characters and objects within the locations given by the 'LocList'. The 'WorldCase' class packages this information so that the case-based planning can effectively compare the current world situation with the library of abstract plot move cases.

The 'StoryCBR' class handles the case based processing and initialises with the 'StoryParser' class, which reads the case library of text files from disk and converts them into a set of 'StoryCase' classes, which are composed of a list of 'StoryFunction' classes. The 'StoryCBR' class outputs the most suitable plot move for execution given the current state of the player character's location and situation. This move is then converted into a 'GoalStack' for execution in the game world.

The 'GoalStack' requires a certain set of roles to be taken, which is represented by the 'Cast' class. A 'Cast' consists of a set of references to game world characters, along with the roles that they are assigned to, from the 9 roles described in previous sections of this thesis. Each move ('GoalStack') will have a 'Cast' object assigned to it, so that if the move becomes dormant, it can be re-activated with the appropriate cast. A currently active 'Cast' list is constantly maintained, according to the characters' social simulation dynamics. The current 'StoryGoal' is assigned to the most appropriate character for the required role, and that character will enact the StoryGoal, creating a 'StoryEvent' which is registered, with the next goal in the GoalStack becoming active. When (a) a GoalStack is complete, (b) the current GoalStack is not making any further progress, or (c) the player moves into a new location with different characters, a new GoalStack is selected using the 'StoryCBR' class to compare the current game world state with the case library.

This SD cycle was implemented to spread the computation of its various activities out over a set time interval. The cycle is divided into sections that can be performed at discrete times. This helps performance, and ensures that the framerate of the game does not suffer at points in the execution when the SD is executing a task such as case selection. The cycle is divided into; (a) getting information about the world state, (b) move selection and combination, (c) move adaptation and (d) move assignation.

5.3. Incremental Testing

Although architectural design was decided early, detailed algorithmic design proceeded along with experimentation with implementation, with three different game worlds being test-beds for different elements of the OPIATE system. The three demonstration games each implement a subset of the features of OPIATE. The first, a simple proof-of-concept demonstration, was called 'Bonji and the Magic Peanut', which used the short script shown in Figure 5.3.

```
villainy Fools! I took their Magic Stone! HAHAHAHAHA ;
mediation find the Magic Stone, I beg you ;
donor give us a hand would you? ;
reciept heres a lucky peanut ;
guidance just use the force ;
struggle you will never beat me! ;
struggle oh no! you have the magic peanut! ;
strugge take the stone, just dont hurt me! ;
mediaion yaay! the Magic Stone's back ;
victory Happy days are here again! ;
```

'Bonji and the Magic Peanut' Script

Figure 5.3: Initial Demo game script

The script was enacted in a flexible enough way, with one character pre-defined as the Villain, the player character as the Hero, and other characters taking the roles of the Donor, the Mediator, and the Helper, depending on the current location of the player. Script lines were enacted by simply displaying the pre-authored dialogue over a character. There was no case-based planning, and no social simulation, and the player had no means of interaction with the system apart from movement around the game world. The only concepts that this implementation proved were that of the usability of the game engine itself, and the fact that different characters could be used to enact plot functions depending on availability in the player character's immediate location.

The second demonstration game developed the game engine, incorporating some social simulation, and featured a multiplayer capability, yet was still lacking the case-based planning component. Based on part of the film 'Star Wars - A New Hope' by Lucasfilm, the player took the role of Luke Skywalker in his quest to rescue Princess Leia. This story was used because it fits very well with Propp's schema, and thus enables an investigation of how flexible the schema can be when integrated with player agency. ActionObjects were not implemented in this game, however, and player agency was thus still relegated to the movement of the player character through locations, reacting to the story events that were portrayed by the various characters. Characters could gossip about meeting other characters, and about story events that they had witnessed. However, the flexibility of the story enactment was still limited to different characters taking different roles depending on availability.

The multiplayer component was tested in this game world, with players taking control of other characters in the game world. The 'ClientManager' class enabled the server to listen for incoming connections from clients, assigning each client a 'WorldConnection' class. This class sent and received messages on both the client and server using UDP sockets. On the client side, messages about the player character's movements and activities were created and sent to the server. On the server side the state of characters and objects in the current location of the player was packaged as a message and sent to the relevant client. The client maintained a 'World' object that was updated via these messages, yet some entities were updated locally, such as the player character.

The third demonstration game, described in chapter 4, implemented all of the features of the OPIATE system, yet the multiplayer component was disabled for volunteer evaluation of the game. The multiplayer component was not developed to the point where confidence in the system was high enough that players could test the system. Also, due to the logistics of evaluation, spread over a number of weeks, it was thought that the small number of volunteers and the availability of LAN access would limit player evaluation of this component.

The evaluation game's environments were created primarily with a single-player story in mind. The logistics of computer game level design for multiplayer and single player are traditionally very different, explained in [Gűttler 2003]. A single-player story tends to lend itself to a progression through a path that is delineated by the environments that the game designer creates, yet a number of successful modern games diverge from this method. Multiplayer environments tend to be very open and allow a lot of freedom of movement. A combination of the two approaches entails designing environments composed of a number of areas that have a very free-roaming character to them, while only allowing player access to the next area after a set of goals have been achieved by the player. This approach has been used by Swartout et al [Swartout et al 2001], in their 'Freeplay Nodes', and is found in the 'GTA' series of computer games, among others.

Propp noted that the journey of the Hero is central to folktales, with his morphology featuring some character functions that force the Hero into new locations, such as the 'Departure', and 'Arrival' functions. Since there are three locales in 'Calabria' (the game world of the third demo game), the player experiences a flexible story in each of the three locales, and can only transfer to the next locale after a puzzle is solved, which is integrated by the SD into a story function. A further logical development of the system would be to base OPIATE in a multiplayer-type environment that allows more complete freedom of navigation around its locales. The storytelling techniques found in games are based around the introduction of new characters, challenges, and environments as a reward for player interaction with the game. In a free-roaming environment, multiplayer or single player, these mechanisms must be represented more abstractly, as the OPIATE system does. This allows a player to freely explore areas of their own choosing, with a storyline emerging based on the available local characters and the player character's interactions with them.

5.4. Integration

This section will discuss how all the components in the OPIATE system work together to present the story to the player in the context of a computer adventure game. An author can choose to define which characters will initially take each of the roles, or these roles can be filled as the social simulation progresses. Thus, either a player character can simply interact with the characters until an appropriate story is selected, or the player is introduced to the characters with an initial story move based on the role selections of the author. Either way, the story develops from the character interactions, and future plot move and role casting selections will be based on how the player has influenced character attitudes.

5.4.1. Authoring

The author's role in creating computer games has come into more prominence in recent times, with a number of successful development teams employing professional writers full-time, to design environments and construct the storyline with the design team. This signifies a growing appreciation of the importance of storytelling to capture an audience's imagination. Steven Spielberg has recently noted the growing maturity of computer games as a storytelling medium [Breznican 2004]. Storytelling techniques that are incorporated into game mechanics tend to be the most effective, and the focus of this work is on the creation of sophisticated game mechanics based on the concept of a player interacting with story mechanics.

As the unfolding of the story is handled by the system, the author's role is to create the setting, the initial situation and the backstory of the characters. Ideally, an author should be able to create the rules that the SD follows in selecting and casting plot moves, but in implementing the OPIATE system, these rules were embedded in the code. These rules effectively represent thematic control of the emergent stories. In a traditional novel, for example, an author uses the plot to elaborate thematic ideas, by portraying a series of events, convincing the audience that these events have some relationship with reality. For authoring a computer game story, the rules of the game mechanics can be the means of elucidating a theme. For example, in the game 'Katamari Damacy' by Sony, the player moves a ball of stuff around, consuming game entities by rolling over them, in order to grow larger and thus becoming able to consume larger entities. It has been noted [Gilbert web] that this game mechanic could be a comment on consumer societies.

5.4.2. Game Mechanics as Story Mechanics

The game mechanics used for the Bonji game are based on two mechanisms; the use of actionObjects, and gossiping. The presentation of character functions is done through graphically animating actions and constructing dialogue describing these two mechanics. The player character is given these same abilities that the NPCs also possess. The player is allowed to control how the player character gossips and uses objects, and the NPCs are directed by the SD to use objects and gossip based on a set of suitable character functions, although these events also take place independently of the SD. These independent events influence the choice of both plot move selection by the case-based planning, and role casting by the rule-based system. Because the plot is changed to suit the dynamics of the game, the interactions that the player can perform become the driving force of the story, and any progress made will be because of something the player has done. This is important for maintaining the player's sense of agency, their perceived ability to effect change in the game world.

A major difference with the OPIATE system and other story-based computer games, however, is not that the driving force of progress in the game is the player's interaction, but that the player's interaction is also the driving force of the direction of progress in the game. Story games are usually experiences 'on rails', much like a ride in a theme park, and this enables a pre-scripted story to be told at certain key points that a player character is guaranteed to pass through. The story is kept rigid, while interactivity and any emergence is seen at a finer grain than the overall story tapestry. The OPIATE story mechanics described in the Design section are implemented at an abstract level, with the story events taking shape in the two key game mechanisms. Story events are gossiped about, and the dynamic creation of character cliques allows a player's story experience to be different each time, even if the same plot move is selected by the SD.

Traditionally, in AI planning applications, it is clear what plans are better for what scenarios, and a heuristic can be created to evaluate this, but for story planning, every story is assumed to be equally valid. Stories differ in their applicability to a given situation, not their success at a constant test mechanism. This philosophy has been used in the case selection strategy of OPIATE. Each move has its own internal structure, and a player negotiates a move by interacting with the events that are created from a combination of the move with the current game world situation. The structure of each move represents a distinct game mechanic, which a player is intended to interpret as a fragment of the overall story. I use the term 'game mechanic' here in a way slightly different to that usually used, both in the game industry and in this thesis.

The game mechanics that each plot move represents in OPIATE are larger in temporal scale than the second-to-second game mechanics governing things like actionObject usage and gossiping. For example, there are larger-scale mechanics in commercial games that govern long-term goals of the player. In Nintendo's Zelda game, The Ocarina of Time, there is a temple which must be reclaimed in the icy hills overlooking the land of Hyrule. When this is achieved, the river thaws and a lake far down on the other side of the gameworld fills with water, allowing the hero to swim across to an island that was previously unreachable. This process comes from the representation of elements in the game such that a chain of events can occur given player interaction. With the Proppian move scheduling, the game mechanics of OPIATE include an abstract representation of story elements arranged temporally, so that these chained events can occur while maintaining the causal links inherent in the game engine. This is why I refer to these game mechanics as story mechanics. The overall story that a player experiences is thus made up of each move, united by their symbiosis with the game world dynamics.

6. Analysis & Evaluation

6.1. Introduction

This chapter is divided into two sections, analysing the system from two different perspectives. The first section will consist of both my own analysis of the system as a model of human storytelling intelligence, and a comparison of it to story-based computer games and to other istory systems that have been produced. The second section will describe the subjective evaluation that took place, and analyse the feedback that was received from people who played the third demonstration game.

6.2. Analysis and Comparisons

In developing the OPIATE system, concepts from both computer games and academic AI were used. Existing computer games have produced atmospheric interactive storytelling that engages players very well. The approaches of game designers are therefore a useful tool in the creation of AI istory systems, yet must be augmented with lessons gleaned from academic projects such as Turner's [Turner 1992].

6.2.1. AI and OPIATE

The SD in OPIATE uses a sophisticated agent architecture incorporating case-based planning, a world view of a complex simulated world, plan adaptation and focused attention for enacting and revising plans, and an expert system that encapsulates knowledge of both the internal relationships in its plans, and how its plans relate to its world view. Goals can be attempted in multiple ways, and new information can be incorporated into plans as they are enacted.

This model is a representation of the thought processes that an oral storyteller goes through in constructing and adapting a story for an audience. Its effectiveness is limited by the complexity of the simulated world that it operates in, where a human storyteller is only limited in the creation of a storyworld by their imagination and breadth of experience. The game world simulation in OPIATE was kept relatively simple, with character modelling that does not improve on previous implementations, both in the games industry and academia. This was a consciously applied limitation to the system, as others are improving technology for complex simulation of characters and worlds, and it was thought that a novel approach to story generation and direction would be a useful addition to such systems. The SD thus comprises the only component of the system that can be regarded as an attempt at the creation of AI. It was assumed that to empower an author to tell true istory, an artificially intelligent system that works as a powerful tool in the storyworld is required. The SD thus represents an author's wishes, and therefore to more fully represent them, the SD must be able to incorporate the desires and goals of that author in its methods of operation.

Because the SD uses a case combination component for the case based planner, the model can be likened to the well-known re-use of story fragments that human authors commonly perform. References to previous stories are to be found everywhere in literature and folktales, from Joyce to 'The Simpsons' by Matt Groening. Utilising other stories in telling new ones is a basic tool of the storyteller's art, and can be utilised for many reasons and in many ways. The limited use of plot move merging in OPIATE is not based on any thematic or novelty criteria, but a simple set of rules that are based on the case selection suitability criteria discussed in chapter 4, along with a set of rules governing Propp's character function groupings. These rules simply enable the SD to replace less suitable versions of functions with those more suitable from another case. Due to the way cases are combined, generated cases can get larger than the source cases, so generated cases are not stored in the case library like the initial cases, but in a separate folder on disk. New cases can only be created with the merging of one or more of the initial case base. If generated cases were used to merge with other cases, a move could eventually be generated with all possible character functions, and it was decided that this was not a useful mechanism. Removal of less suitable functions from cases was considered to account for this, but was not implemented, as the structural integrity of the cases could be compromised. Less suitable functions could serve as elements to unite other functions, so it was deemed that it could be detrimental to the plot to allow them to be removed.

The SD implemented for OPIATE did not have explicit models of author goals, but these were manipulated in the coding of the system. The tools that should be available to an author in creating a storyworld are thus not available as an author interface in OPIATE, but as an academic project, this work explores the tools that could be made available through the introduction of the SD as a mediator of a story author's wishes. This was achieved by allowing the SD to operate without explicit goals, allowing it to pursue the direction of plots that were found to be suitable to the game world situation. Propp's classification scheme was used as a resource by the SD, yet because this scheme is independent of theme, there was no way to include thematic criteria in the selection of plot moves. This, in fact, enables the OPIATE architecture to be applicable in many types of storyworld, with artistic themes authored in each storyworld individually. Thematic knowledge and authorial control of themes should thus be implemented at the level of enactment; the casting of characters in roles and the simulation and interaction rules of the storyworld.

The success of the system is thus debatable, as the OPIATE architecture does not directly solve the problems that surround authorial expression in complex simulated worlds. What it does achieve is the possibility of structuring events in such a world such that they appear in the form of a coherent plot, no matter what events the dynamics of the world produces. A required partner to OPIATE is thus a thematically simulated game world that is open-ended and free-roaming for the player. Progress in the creation of simulated game worlds in commercial games is limited by the increasingly hit-driven marketplace and the restrictive role that big-name publishers have in the creation of new concepts for game mechanics. Academic projects have a degree of freedom from commercial concerns that enables projects to pursue ideas that may not have immediate value to a marketplace, and this has enabled the OPIATE system to be developed as an experiment in AI storytelling. The limitation that OPIATE's useful area of application is in thematically simulated game worlds means that the technology will only show its true potential if it is incorporated in the simulated worlds that are found in such games as 'The Sims', 'Fable', and 'Thief'. An analysis of the SD architecture is desirable, to evaluate how it could be incorporated in such game worlds.

Because the SD agent bases both case selection and casting decisions on the dynamically changing attitudes of characters, an incorporation of OPIATE into another game engine would only be feasible if this attitude modelling were already in place to some degree in the existing engine. Simulation games can exist without the plot planning activities such as those found in OPIATE, but OPIATE could not operate without such a simulated game world. Since 1990, the research focus of cutting-edge AI has been on social models as well as cognitive models, and the situations that an AI implementation is faced with are less restricted, allowing emergence of unplanned situations to influence an AI system's learning. From this view, placing an AI agent within a complex simulated world is an approach that is more promising for the development of truly adaptive and intelligent systems than feeding it a corpus of pre-rendered data. Because the OPIATE SD was only fully implemented in the third demo game, it was impossible to test its functionality over a range of environments, but in playtesting the game, surprising situations were sometimes observed.

As the social simulation proceeds, characters form groups and gossip is spread through groups, but inter-group gossip is also possible. The player character, character A, can interact with a character B by performing an action that positively effects the attitude of character B towards character A, but this can then be gossiped about and a character C that has a negative attitude towards character B can then develop a negative attitude towards character A. The SD can then assign a villainy function, or some other character function requiring a negative attitude for the player character, to character C. Thus, through the player performing a positive act, a negative act can be initiated by the SD. This is an example of an artefact of the social simulation that was not explicitly designed, and other more complex situations can be seen to emerge from the simulation.

The required separation between the SD and the game world is not as desirable as it first seemed in designing the system, and to be useful in a new game environment, the SD as it is currently must be re-implemented to fit the interactions and mechanics of such an environment. For the inclusion of an SD agent in other games, it is required that designers are allowed access to the SD algorithms so that they can customise them. This is one of the improvements to the system that will be included in the 'Future Work' section of the next chapter.

6.2.2. Computer Games and OPIATE

The emergent storytelling found in games such as the 'Thief' series, among others, is comparable to that found in the social simulation of OPIATE, yet because the plot is not orchestrated by an author, this emergence is allowed to dictate the unfolding of a higher-level plot. The gameplay mechanics that were designed for the testbed games are based on the inventory and exploration systems of adventure games. Ron Gilbert, creator of the 'Monkey Island' series, has said that he thinks simulation can create more immersive game worlds for adventure games, and his original plans for 'The Secret of Monkey Island' included a more fully simulated social system, with characters being modelled with their own goals [Gilbert web]. This was cut from the game because at the time it was too complex to implement within the limitations, but current adventure games have not taken advantage of the possibilities of simulation that Gilbert recognised early. The paradigms of adventure game storytelling are made more complex by introducing social simulation, but a system such as OPIATE could be used to provide more abstract plot control, overcoming some of these problems.

The adventure game 'Discworld Noir' used one of the most sophisticated storytelling approaches of any game [Hobo 2001]. Story events were painstaking created in separate threads that could be enacted in a flexible order, with key events marking the opening of new areas of exploration. The standard inventory system of adventure games was extended to include 'clues' that could be combined and used like other objects, which was a vital part of the story that emerged in the player's mind. When clues were combined in the right ways, the main character realised connections that might trigger new events in the story. This is the sort of game mechanic that could be utilised to great effect with an istory system. OPIATE uses a similar, albeit more limited, approach to game play, with gossip events being a key part of a character's inventory. The use of this gossip inventory can trigger new attitudes and facilitate story events to be scheduled based on those attitudes. The gossip system used in Chris Crawford's 'Trust and Betrayal' game was an early example of the possibilities of social simulation in games, and although I haven't played this one, Crawford's trademark complexity was used. Crawford's extensive writings can be found on his Erasmatazz website [Crawford web].

One of the main features of Propp's plot scheme is that it is independent of dialogue and is based on the conception of the plot as a series of events. The dialogue generation of OPIATE is very simple, and conversations are not scripted. This contrasts starkly with the painstakingly crafted scripting and recording of dialogue that is necessary for modern story-based computer games. This rigidly pre-scripted approach can be used for flexible storytelling, if story events are designed so that they can happen in a different order depending on the player's activity, but to enable true interactivity in the plot, story events must be created with a more abstract framework.

6.2.3. Istory Systems and OPIATE

The use of simulation has been used in relatively few istory systems, with the model of storytelling being limited by the conception of story being in direct opposition to player freedom. Crawford's personality and emotional simulation is the most extensive, and he uses personality traits, with shorter term mood traits, and a large range of verbs that dictate how moods will be dynamically altered. On Mateas's scale from 0 to 100 described in the Background section, where 0 is fully autonomous characters, and 100 is a story fully controlled by a central mechanism, the OPIATE system would fall at roughly 50, while Crawford's would lie lower on the scale, at perhaps 20 to 30. Any central control of Crawford's Erasmotron stories is done by the use of the author toolkit in defining interactions. Every time a story is experienced, the use of random variables allows situations to occur with different characters, but because certain verbs are defined by the author as thematically important, those types of interactions will occur in the story. Cavazza and other autonomous character-based approaches are near 0 in this scale, with fully autonomous characters interacting to create events.

Other istory systems, by contrast, can be classified much higher on Mateas's scale. The approach of Sgouros [Sgouros 1999] uses the concept of creating and resolving conflict to push all the character interactions, while Sgouros sees a story as a set of processes that act on characters to force them into providing and removing obstacles. The use of simulation is often seen as being fundamentally opposed to the use of temporally sequenced plots, and Wright sees the two as the difference between control exercised by the player and control exercised by the author [Wright 2003]. However, with the OPIATE system I have shown that the two can work at different levels, with plot-level simulation taking place at a level more abstract than world simulation, with a feedback between the two that ensures consistency.

The choice of Propp's folktale scheme as the central story model meant that the system would be most suited to storyworlds that are not fully realistic in their depiction of characters. Stern’s approach for the Façade system used a comic-book-like depiction of characters and this is a concession to the phenomenon of 'the uncanny valley ' that Masahiro Mori observed in people's reactions to almost fully realistic character representations [Mori 1982]. People react poorly to characters that are depicted in their appearance as being realistic humans, if their behaviours are simplistic and unrealistic. Thus a cartoon approach was taken to create the demonstration storyworlds, so that they would fit with the simple character simulation and folktale plots.

6.3. Subjective Analysis

Subjective analysis proceeded by requesting a variety of people to download and play the demo game, and fill in a web questionnaire about their experience with the game. The game was presented without giving the users any foreknowledge of the game, allowing them to explore the gameworld as they saw fit. Some questions were about the player's previous experience with computer games and their preferences, to help elucidate any inconsistencies in results.

6.3.1. The OPIATE Questionnaire

The questionnaire can be found in appendix D, with all the results. Results were not consistent over all respondents, but this was anticipated due to the subjective nature of both story and game evaluation. Respondents were thus asked how often they play computer games, from 1 to 10, and were also asked about their preferences of story. It was assumed that stories lie in a three-dimensional space, and respondents were asked to provide a preference that lies at a particular point in this space. One dimension goes from 'action' to 'drama', the second from 'movies' to 'novels', and the third from 'romantic' to 'dark'. This was assumed to meet the need of differentiating between people with wildly different responses to OPIATE, but nevertheless, inconsistencies remained.

There were three sections to the questionnaire. Section one deals with the overall experience of the player and how much the experience was seen as a story. Section two relates to the believability and consistency of the game world and how the characters came across. Section three is about the player's own preferences and how they compared OPIATE to other computer games.

6.3.2. Results and Analysis

The 3D story classification space was useful for differentiating some answers, most notably question three: 'Did the game experience as a whole seem to you to be in the form of a story?'. This question received the most positive response. Respondents who prefer action stories universally responded positively to this question, with drama fans rating this 'storyness' lower, particularly those who answered in favour of novels over movies. One noteworthy result was that those whose preferences lie in the centre of the classification space rated the 'storyness' of their experience the lowest. This is where those who rated novels and movies equally, action and drama equally, and dark and romantic stories equally, are characterised (Figure 6.1).

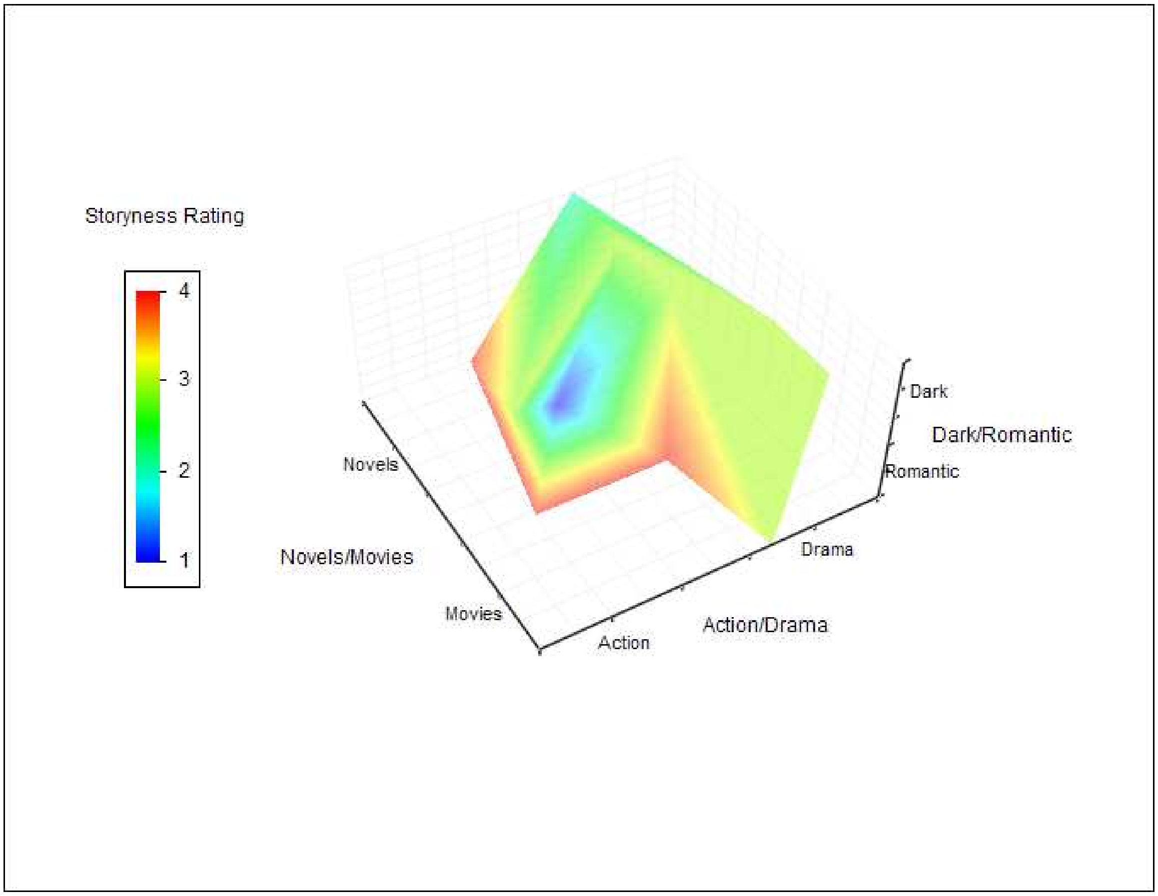


Figure 6.1: Rated 'Storyness' against Players' Story Preference

OPIATE's output has been evaluated as a story by those players who prefer action-based storytelling, and this corresponds well with the approach taken, and Propp's action-focused scheme. Fans of both dark and romantic stories, and both novels and movies responded well to OPIATE. Only fourteen respondents were involved in the evaluation, so the central anomalous negative rating could be partly due to this statistically unviable nature of the evaluation. However, it could be argued that the result can be explained by the unusual nature of the storytelling involved with OPIATE, and hence its incompatibility with a conception of stories that is more middle-of-the-road.

Figure 6.2 shows that those who thought their experience was more story-like enjoyed playing the game more. This confirms that players engaged with the system as if it was a story, and correlated the game mechanics with an interaction with a story.

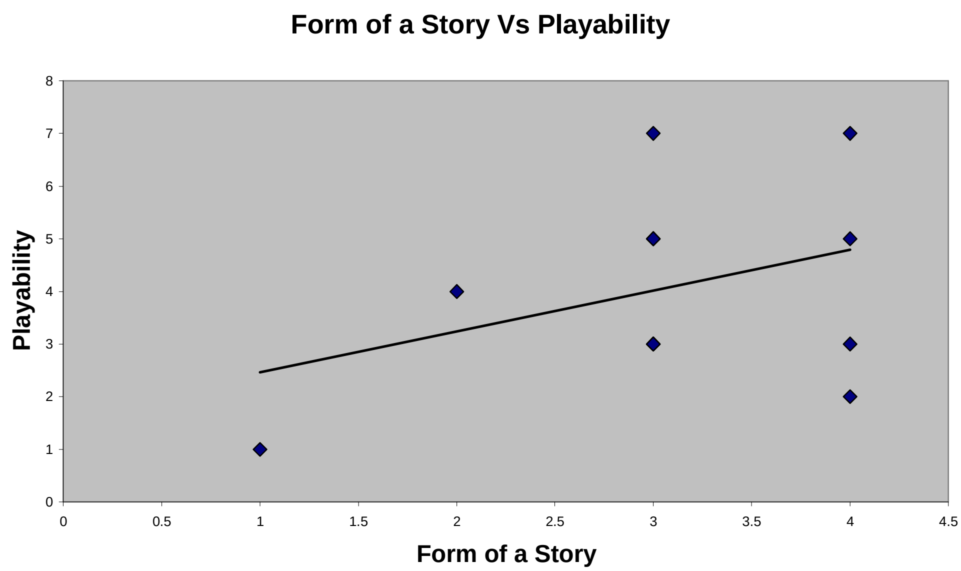


Figure 6.2: Graph I

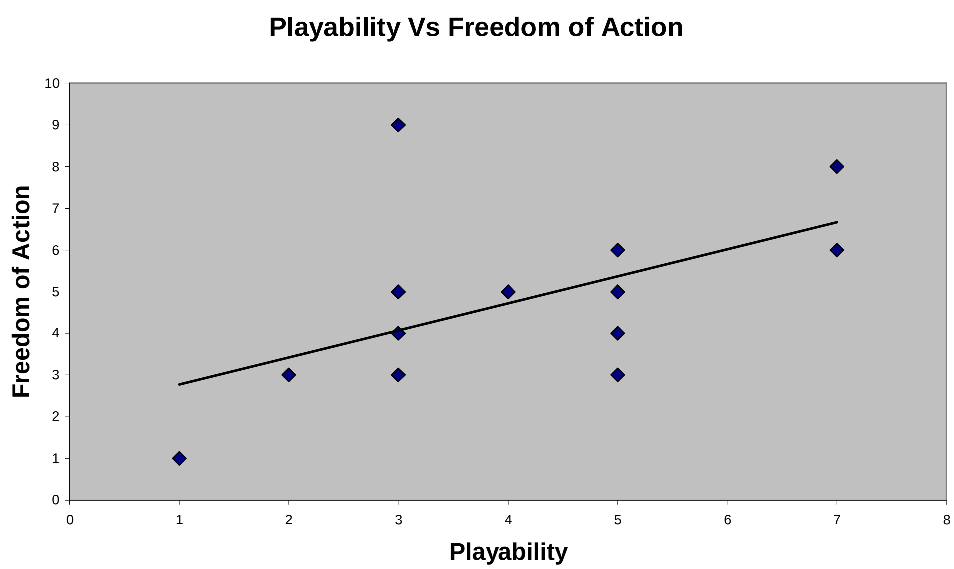


Figure 6.3: Graph II

Figure 6.3 shows that those who felt a freedom of action in their interaction with the system also enjoyed playing the game. This shows a stronger correlation than the previous graph, and makes sense given the assumption that players enjoy computer games because of the freedom that they are afforded in the game world. Respondents who played games the most often rated the playability of OPIATE in a variety of ways, and this graph is inconclusive and is not shown, but this 'often play' value differentiated the perceived believability of the game world rather better (Figure 6.4). Players who are used to the more sophisticated game world simulations found in some modern games were less prepared to suspend disbelief for their interactions with OPIATE, although overall the believability ratings are high.

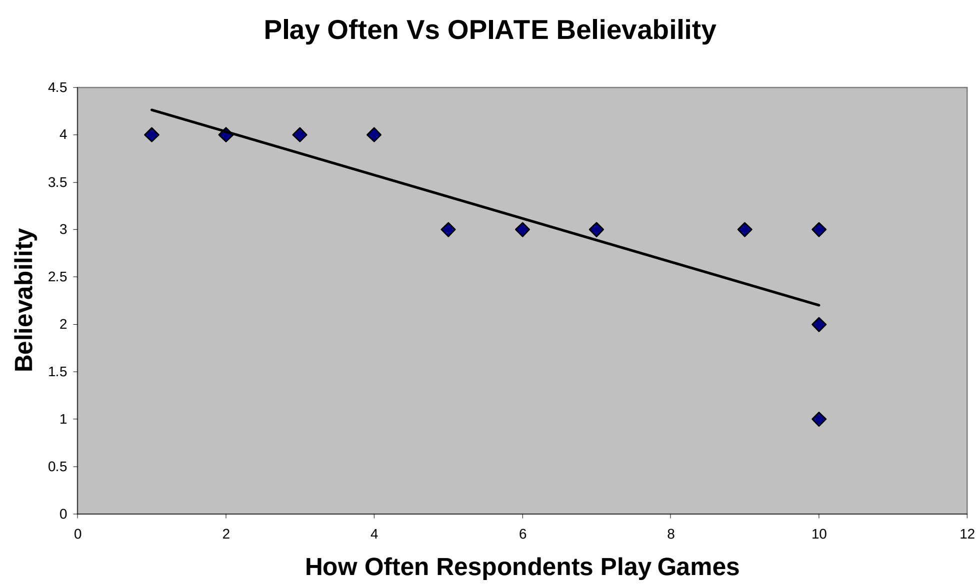


Figure 6.4: Graph III

Figure 6.5 shows the plot of freedom of action vs. the influence that a player felt they had on the unfolding of the story. Respondents who felt they had freedom of action in the game world also felt that their actions influenced the unfolding of the story. This is a strong indication that when the system works, it achieves the basic goal of presenting a story as an interactive system.

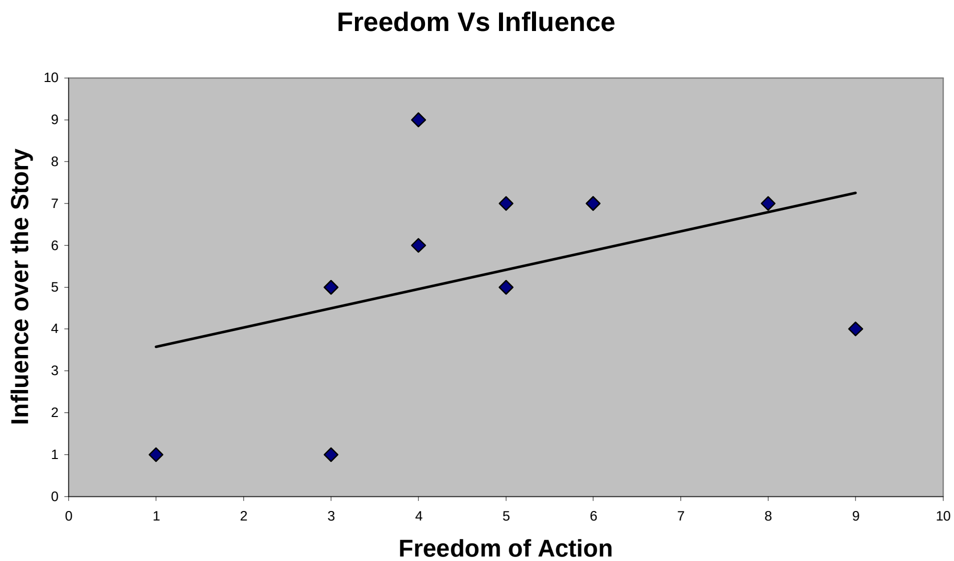


Figure 6.5: Graph IV

Those who felt they had freedom of action also felt that the game world was more believable (Figure 6.6). This shows a stronger correlation, as the two variables have more to do with the game world interactions than a link between game world and story considerations. Freedom of action was a stronger differentiator than the freedom of movement question, although those who felt a freedom of movement also tended to feel a freedom of action.

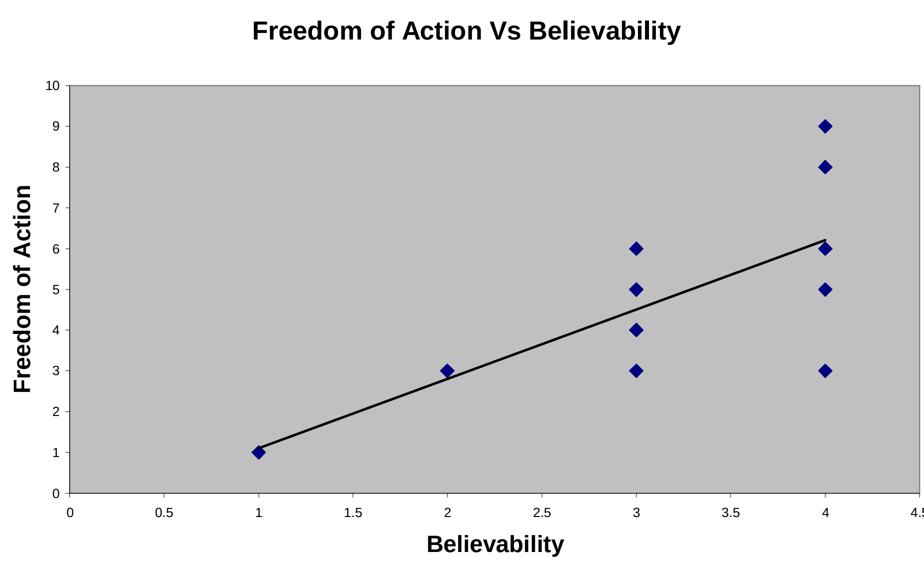


Figure 6.6: Graph V

The results shown in the last few graphs would seem to suggest that those who play games more often might respond negatively to the system, although this is not seen in a direct correlation. The demo game is not goal-oriented like other story games are, and therefore players reacted based on their interaction with the system as a story, instead of a game, independent of their different levels of gaming experience.

These results taken as a whole are a positive reaction to the system, vindicating some of the assumptions made in design, and illuminating some problems that arose as a result of the implementation. Comments consistently rated the system down in the areas of easiness of play, and character complexity. A number of players said that they did not know what they 'had' to do next, which shows that people assumed that they were on a set course through the environment, yet felt they had an influence over the story. This was a basic working assumption, and is the chief distinction that the medium of computer games has to offer over other forms of storytelling media. A story must be authored as the creation of a world with believable characters, allowing incremental player progression through the world. Propp's centrality of the journey means that a story must be a progression. Whether it is an emotional progression, as found in drama type stories, or a physical or material progression, is genre issue, but in either case a world where exploration and interaction can take place is a welcome addition to the complexity of the storyteller's art.

7. Conclusions and Future Work

7.1. Introduction

The OPIATE system is one of a number of projects currently investigating the use of AI techniques for storytelling on computer. The application focus is on the applicability of these techniques to commercial computer games. This thesis has laid out a methodological approach to solving the problem of creating 'istory', the concept introduced in the introduction chapter. Allowing a player to have as much influence over the plot of an interactive story as possible was facilitated using a combination of paradigms from both academic AI research and computer game development. A story director agent was used to dynamically and adaptively plot a story that is experienced by the player through interaction with a simulated game world.

This chapter will present conclusions that were drawn based on both the assessed suitability of the approach, and the evaluated response of players. Then a range of possible future improvements and additions to the system will be discussed, and a final comment will be made concerning technologically assisted artistic expression.

7.2. Thesis Summary

7.2.1. The Combination of Paradigms

The suitability of Propp's scheme and case-based planning to modern story games is in question, and from the implementation of the OPIATE system, some conclusions have been drawn. These will be borne out by the evaluation conclusions.

- Propp's scheme proved eminently suitable for use in a case-based system, as his skeletal plot templates match with recent developments in case-based planning concerning skeletal cases, and local adaptation.

- Propp's scheme is also useful for action-based computer games, and his functions have correlates in game-play mechanics of existing genres of games.

- The assumption that Propp's scheme can lend itself to player agency has been corroborated, and three key methods for integrating player agency in the plot were designed and evaluated. These are in the move **selection**, **casting** and **initiation** mechanisms.

- Case-based planning was found to be an effective real-time planning paradigm for use in a simulated game world, with many common CBR problems alleviated by the relatively small case library, provided by Propp's corpus.

- The division between plot-centric and character-centric istory telling was bridged by allowing the plot planning to work on a higher level than the character simulation, but maintaining consistency between the dynamic simulation and planning using adaptive casting.

7.2.2. Interactive Simulation Facilitating Collaborative Imagination

The collaboration between an author and an audience has recently been enabled through the use of computers as a storytelling medium. Since this collaboration has always existed in the form of oral storytelling, people do not see it as an intrinsically foreign concept. In relatively modern story media such as the novel, television, and movies, this collaborative mode of storytelling has lost its hold, and a growing number of people from mainstream culture are seeing that interactive computer entertainment has a novel power to hold their attention.

Although in its infancy, the computer game industry is having an increasing effect on technology and culture. Like the movie industry in the early twentieth century, the industry is currently led by technologists who have the technical skill to keep new products interesting. A number of key game designers that have skill in both artistic and technical areas have proved instrumental for the acceptance of games in wider culture, but the divide between artistic and technological industry staff is an oft-quoted problem for games studios. With graphics and animation, the current norm is to make tools that artists can use to create content for the game worlds. This process has gone far enough in the movie industry to accommodate standards and common practices that differ from the approach used in the game industry, where sets of tools and methodologies often have to be created for each individual game project, but games are seeing a new surge in middleware tools that enable the increasingly complex gameworlds to be created with as much ease as the less complex games from yesteryear.

The introduction of storytelling tools to this ontology will allow more traditional story authors to exercise control over the interactive experiences of players, but established game designers, like Will Wright, have already recognised the potential of allowing freedom in their imagined game environments, while exercising control over the reactive behaviour of game systems. With the OPIATE system, I have explored the use of social simulation to create an artificial society where situations can emerge based on player interactions. The creation of a game world in OPIATE is an exercise that does not constrain a player to a specific path, but allows them to create their own experience through interaction with the rules and dynamics of the system. This is a powerful metaphor for artistic expression that I believe will become more respected in the rest of the twenty-first century.

7.2.3. Interactive Plot as Thematic Conversation

The use of thematic simulation has been used in relatively few games, but with the development of flexible author toolkits, the design of custom simulation systems will allow authors to share control with each player individually and allow a computer-mediated 'conversation' to occur based on the rules of interaction that an author sets up. The imagined game world of an author is entered by players, who choose how they interact with the system, and create their own view of the author's world. This has already been achieved by some game developers, but the creation of tools for authoring thematic systems explicitly will allow non-technical artists to explore issues of their choosing with a player who can decide how to interact with the author's set of themes. The introduction of plot-level simulation into this hypothetical author toolkit is a possible extension of the author's thematic reach. Authoring themes implicitly as the simulation rules of a game world is one possible level of author communication, but the potential for authoring a dramatically interesting story is limited by the freedom that the player has, combined with the lack of higher-level simulation rules that deal with the temporal structuring of events in the world.

Propp's morphology has been utilised in the OPIATE system to provide a structuring mechanism for the game world events. This does not provide a direct authorial control over the themes of the plot, but the rules and case based selection mechanisms are a set of tools that provide an implicit control analogous to the control exercised in the design of world simulation rules. As no other authors created story worlds for OPIATE, this was an acceptable way to test the tools and experiment with selection and casting mechanisms. The manipulation of these mechanisms represents the proposition of a set of constraints that surround a space wherein all possible stories that can emerge from player interaction are defined. This is a complicated way of defining the consequences of a player character's actions in the game world, but allows both author control over the high-level thematic plotting while affording the player considerable freedom of movement and action in the game world. The author is effectively implicitly defining a set of themes, allowing the player to interact with those themes, and instructing the system to respond with a story plot that presents those themes. Thus, within a certain limited framework, a conversation occurs between the player and the author's representative, the story director agent.

7.3. Implementation & Evaluation Conclusions

7.3.1. Implementation Conclusions

The general approach of organic design and implementation was successful in that it allowed an incremental investigation into the characteristics of the architecture.

The game world and story director agent were developed in tandem, with one growing more complex alongside the other, to ensure that they functioned properly. The use of C++ was necessary, and provided an efficient implementation that could update all 28 socially simulated characters along with the story director logic while maintaining a steady framerate of about 50 frames a second. Experimental environments were tried, and it was found that the social simulation (without SD logic) could operate with over 300 characters being updated simultaneously. The story director agent was implemented with a sophisticated agent architecture, and operates in the dynamic simulated world, adapting and re-planning when necessary.

There were a number of limitations that were encountered in the OPIATE implementation, yet the overall architecture was successful. In the multiplayer version, network lag and client/server inconsistency of a real-time world was a problem, but this was due to insufficient time and resources for a professional-level multiplayer game engine. There were also limitations in the simplistic character modelling approach, yet this was a conscious decision that was based on the focused nature of the research into story direction technology.

7.3.2. Evaluation Conclusions

Analysis of the feedback from the respondents to the questionnaire website provided positive confirmation of a number of basic assumptions that were made in the design of the system.

- There was an average 79% response to the question 'Did the game experience as a whole seem to you to be in the form of a story?', which validates the assumption that the system's dynamics result in an aesthetically story-like experience.

- Respondents who rated action stories above drama stories rated the 'storyness' of their experience higher than the others. This shows that the action-based storytelling methodology was consistent with the results.

- The simplified character modelling resulted in veteran game-players' recognising the relatively inferior believability of the game world, although believability was rated high relative to other criteria, such as ease of use. The social simulation served its purpose well enough to carry the storytelling.

- Respondents who felt they had freedom of interaction retained a feeling of influence over the unfolding of the story. This verifies the basic assumption that these two characteristics are fundamentally *not* mutually exclusive.

7.3.3. Limitations of the Evaluation Scheme

The most obvious limitation was that only thirteen people filled out the questionnaire. People may have been put off by the time commitment (20 to 40 minutes) and the fact that they had to download and install a program, with the prevalent caution about computer viruses. Some people could not get OPIATE to run on their system, and some played the game without responding to the questionnaire. Although the results are not technically viable for rigourous statistical analysis, they corroborated some of my own analysis.

The evaluation took place with a single version of evaluation software, and to more scientifically test the characteristics of the architecture, each component should be tested independently. Respondents were encouraged to try the system with the 'turn off story AI' option activated, but the results of this were inconclusive, as many did not, or did not do so for a long enough time.

7.4. Future Work

7.4.1. Further Evaluation and Development

The system as it is currently implemented is flexible enough for a different type of game world to be accommodated. The environment created for the evaluation consisted of three areas that the player character progressed through with a high level of freedom within each area, but a fully open and free-roaming environment would facilitate an evaluation of the ability of the system to portray a story journey while the player is given full freedom of movement. A more extensively authored game world, with a wider range of actionObjects available, would allow more freedom of action, and the SD's flexibility could be further evaluated.

A more efficient messaging system could enable better networked multiplayer performance, and the SD implementation could be extended further to enable some client-side story direction, while answering to a master SD server. The multiplayer component is promising, because with the current architecture a single player can experience a story in the game world, and with multiple player characters, the social simulation execution is not made more complex. Because the character attitudes are an artefact of the simulation, even more complex situations could emerge from multiple player interaction with the NPC social simulation system. The direction of a separate, interwoven story for each player is eminently possible with the OPIATE system, and could result in a completely new genre of multiplayer game.

7.4.2. Increased Author Control

For the system to represent an author's wishes, the case selection and adaptation rules should be made as flexible to an author as possible. The representation of these rules as scripts would make possible a higher-level control of the system that would allow author-defined consequences for player interactions, while maintaining the emergent quality of story events. This component is vital for OPIATE's applicability to other game worlds.

7.4.3. Other Game Environments

Although the games that were developed were based on the graphic adventure genre, the application of OPIATE to other games would necessitate a broadening of the scope of the system. Because the SD bases planning decisions on a dynamic 'world view' of the game world, the implementation of the world view classes should be extended so that they can deliver pertinent information about character attitudes and object availability within other types of game. The inclusion of an AI adaptive perception component could be necessary for the SD to automatically adapt to the type of interactions and gameplay that takes place in each new game world. Ideally, the SD should not have to be re-implemented for each new game, but if a game world was defined using concepts similar to the attitudes and inventory-based interactions used in this project, it would be easier to incorporate the SD technology. Game worlds in current popular games are usually limited to simple interactions such as these, but OPIATE should scale to more complex character models if the SD case rules were extended alongside them.

Although Propp's scheme and the vast majority of computer games are action-based, this is not necessarily an intrinsic limitation of the OPIATE architecture. Minsky's 'Society of Mind' [Minsky 1987] proposes that conscious minds are made up of societies of mindless components that together exhibit the qualities we associate with intelligence and consciousness. To extrapolate this metaphor, the NPC agents that take Propp's roles in OPIATE could each be modelled as a component of a single, more complex, character which could undergo conflicting urges and drives. This would enable a more character-focused, less action-oriented, set of game mechanics to provide a complex system in which Proppian plotting could take place. As briefly explored with the use of the 'problem' actionObject in OPIATE, character interactions could be redesigned to rely less on physical objects, but emotional or desire-based components to facilitate this.

7.4.4. Author GUI and Toolkit

The possibility of creating story-direction middleware technology for use by the games industry became a background goal of this project when it was realised that the tools had to be fully crafted before a toolkit could be put together. With the tools still in an evaluation stage, the improvements mentioned in the previous two future work sections must be addressed before a game genre-independent set of tools could be created.

Once this is done, however, a set of C++ classes, along with a GUI for author creation of story rules, would be a promising addition to the games middleware market, and could enable a new level of simulation to enhance story-based gameplay. This would proceed by allowing the scripted plot scripts to be edited alongside the SD rules that govern:

- Case selection
- Case adaptation
- World view formation

It would be necessary to use one integrated Interface for all this, to ensure that rules were consistent. If a case selection rule was altered, it would change how effectively rules in the other two categories operate. This is a requirement that would ensure the story director makes decisions based on a consistent knowledge base. The cases that were created for OPIATE are adequate until new types of character function are required by an author, and if an author wants to do this, a deeper set of tools for accounting for new function types would be needed. New categories of rules would be necessary, and I would recommend a more sophisticated AI architecture for further extension of the SD.

7.4.5. Fractally Harmonic Super-plots

Briefly mentioned in the design chapter, the use of self-similarity that extends outward to larger structural features could enable the SD to simulate a longer story plot based on player interaction. The comparison of story composition to fractal structures is limited, but Laurel's discussion of Aristotle's 'Pattern' layer indicates that lower-level story elements can share structural features with a higher level to enhance the 'pleasurable melody' that is experienced by an audience. Longer story arcs that mimic the more immediate situations of a player would be a reverse of the usual use of Aristotelian pattern, but would maintain the end result of patterning small scale elements on larger scale features.

Some integer variables could be experimented with, such as the maximum depth of self-similarity, and the number of low-level plot moves that are merged to contribute to the structure of the higher level plot. These mechanics could also allow a multiplayer implementation to automatically structure a high level story arc based on the combination of each player's ongoing story.

7.4.6. Storytelling Society Simulation

The gossip system implemented for OPIATE is not as complex as that designed by Crawford in 'Trust & Betrayal', yet is the key to increased believability for characters. Socially simulated characters have almost limitless room for improvement, and are a more direct way to create believability than purely cognitively simulated NPCs, which must incorporate explicit models of social interactions before social behaviour can be observed.

An extension of the gossip system that incorporates Propp-based story morphologies would enable a tighter symbiotic bond between characters and an omniscient SD agent. Characters could gossip about events based on their own experience, and internally structure events to fit the character function scheme, to create gossip messages that take the form of whole plot moves. Each character could have a concept of themselves as the hero of their own story, and effectively become an AI story-teller while participating as a character in a story. Models of emotion, mood, and personality could augment this architecture to portray convincing characters for next-generation interactive entertainment.

7.5. Some Final Thoughts

As virtual worlds become more and more complex, asymptotically like the infamous 'Matrix' from the eponymous Warner Bros. film, simulation and algorithmic generation of the content of these worlds will become more feasible than explicitly designing it. To tell a story in this sort of virtual world would require an ontology of storytelling which derives from observed behaviour of stories in their natural habitat of human culture. The principles of this ontology would need to encompass notions of genre, theme, drama, plot, and tone, and it would need lower level tools to control the specifics of the story to allow the personal style of the artist the fullest expression possible.

The personal touch of an artist is necessary for the satisfying communication of a story, and in a simulated world this requirement is even truer. The author must work at a more abstract level, not directly writing all the dialogue, but using their vision to create convincing character personality models, explicitly defining their inner drives and idiosyncrasies. The world must be fully fleshed out and the means for the desired themes to shape it must be designed, so that an abstract model of the plot can find its expression in the characters.

In order for an istory system to be truly independent of the context in which a story is to be told, models from diverse scientific disciplines, from linguistics and social science to psychology, would need to be implemented in a sophisticated character and society simulation, alongside an AI author representative equipped with expert knowledge on themes, tones, plot and many other areas. The creation of a truly human-level AI is not an end in itself for me, but the problems faced by storytelling for interactive games will require such advanced technology for their realisation, that it is likely to seem like a human-level intelligence. I think that it is a remarkable thing that so many important areas of human endeavour converge in a pursuit that is, in essence, useless, that of playing games. Perhaps it is the things that we do for no reason that ultimately make us who we are.

Not such the World's illusive shows;
Her wingless flutterings,
Her blossoms which, though shed, outbrave
The floweret as it springs,
For the undecieved, smile as they may,
Are melancholy things:
But gentle Nature plays her part
With ever-varying wiles,
And transient feignings with plain truth
So well she reconciles,
That those fond Idlers most are pleased
Whom oftenest she beguiles.

- William Wordsworth, 1832

References

[Aarne and Thompson 1964] Antti Aarne & Stith Thompson, *The Types of the Folktale*,Second revision. Helsinki 1964

[Aarseth 1997] Aarseth E. J., *Cybertext: Perspectives on Ergodic Literature,* The Johns Hopkins University Press 1997

[Abbott 2002] Abbott, W. T., *White Knowledge and the Cauldron of Story: The Use of Allusion in Terry Pratchett's Discworld* Masters Thesis East Tennessee State University2002

[Adams 2001] Adams, E., "Will Computer Games Ever Be A Legitimate Art Form?" in Proceedings of the GAME DEVELOPERS' CONFERENCE , 2001

[Andrews 1915] Andrews, C., *The Technique Of Play Writing* The writer's library 1915

[Aristotle 50BC] Aristotle, *Poetics* 50BC

[Afanas'ev 1976] Afanas'ev, A, *Russian Fairy Tales* Random House 1976

[Azuaje 2002] Azuaje, F., "A Computational Evolutionary Approach to Evolving Game Stragey and Cooperation", *IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 32, No. 5*, 2002

[Badman 2003] Badman, *Randomize* Entry on 9-23-03, 2003. Available online at: http://darkbox.pentagod.com/

[Barad and Robertson 2000] Barad, J., Robertson, E., *The Ethics of Star Trek* HarperCollins Publishers; 1st ed edition 2000

[Barr and Rewari 1994] Barr, A., Rewari, A., "AI in Customer Service and Support, Including Help Desks", *The Twelfth National Conference on Artificial Intelligence, Sixth Annual Conference on Innovative Applications of Artificial Intelligence* IAAI 1994

[Barthes 1957] Barthes, R., *Mythologies*, Paris, Seuil, 1957

[Barthes 1993] Barthes, R., *A Roland Barthes Reader*, Vintage, London 1993

[Bates 1992] Bates, J. "Virtual Reality, Art, and Entertainment" *Presence: The Journal of Teleoperators and Virtual Environments*, 1992

[Bates 1994] Bates, J., 1994, "The Role of Emotion in Believable Agents", *Communications of the AC*M. 7 (37): 122-125 1994

[Bates et al 1992] Bates, J., Loyall, A. B., Reilly, W. S., "An Architecture for Action, Emotion, and Social Behavior" Techical Report CMU-CS-92-144, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. May 1992 Available at: http://www2.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/papers.html

[Bates et al 1992b] Bates, J., Loyall, A. B., Reilly, W. S., "Integrating Reactivity, Goals, and Emotion in a Broad Agent", Techical Report CMU-CS-92-142 1992

[Bentley 1999] Bentley, P., *Evolutionary Design by Computers*, Morgan Kaufmann 1999

[Bento et al 2002] Bento, C., Cardoso, a., Wiggins, G. A., (Eds.) *Approaches to Creativity in Artificial Intelligence and Cognitive* 2nd. Workshop on Creative SystemsLyon, France, 22-23 July 2002

[Bergmann 1991] Bergmann, R., "Knowledge Acquisition by Generating Skeletal Plans from Real World Cases", *Contemporary Knowledge Engineering and Cognition* 1991

[Bergmann et al 2003] Bergmann, R., Althoff, K.-D., Breen, S., Göker, M., Manago, M., Traphöner, R., Wess, S., "Developing Industrial Case-Based Reasoning Applications -The INRECA Methodology" *Series : Lecture Notes in Computer Science 2nd ed.*, Springer 2003

[Berner 1998] Berner, E. S., *Clinical Decision Support Systems: Theory and Practice*, Springer-Verlag; 1st edition, 1998

[Braun 2002] Braun, N. “Narrative Semiotics for Computer Games”, *in Challenge of Computer Games*, Lodz, Poland, 2002

[Braun 2002b] Braun, N. “Automated Narration - the Path to Interactive Storytelling” *in Workshop on Narrative and Interactive Learning Environments*, Edinburgh, Scotland, 2002

[Braun 2003] Braun, N., “Storytelling in Collaborative Augmented Reality Environments” *In: Skala, Vaclav; WSCG 2003. Conference Proceedings*, Plzen, University of West Bohemia, 2003

[Braun and Schneider 2003] Braun, N., Schneider, O., "Suspenseful User Experiences in Collaborative Virtual Spaces*" in Conference Proceedings of HCI International* 2003

[Bremond 1974] Bremond, C. *Logique du récit*. Seuil, 1974.

[Bremond 1995] Bremond, C., Ed., *Thematics - New Appoaches* State University of New York Press

[Breznican 2004] Breznican, A., Interview with Steven Spielberg and Robert Zemekis on USA today, available online at: http://www.usatoday.com/life/movies/news/2004-09-16-game-movie-meld_x.htm 2004

[Bringsjord and Ferucci 1999] Bringsjord, S. and Ferucci, D., *Artificial Intelligence and Literary Creativity: Inside the Mind of Brutus, A Storytelling Machine*. Laurence Erlbaum, Mahwah, NJ. 1999

[Campbell 1949] Campbell, J., *The Hero with a Thousand Faces*, Princeton University Press 1949

[Canham and Tyrell 2002] Canham, R.O., Tyrrell, A.M.,"A MULTILAYERED IMMUNE SYSTEM FOR HARDWARE FAULT TOLERANCE WITHIN AN EMBRYONIC ARRAY" in Proceedings of 1st International Conference on Artificial Immune Systems, 2002

[Carmack 2004] Carmack, J., *Keynote lecture in Proceedings of the Game Developers Conference* 2004

[Carter 2001] Carter, S., and Carter, D.,"Organic Development Process" in *Proceedings of GDCE* 2001

[Chinh 1997] Chinh, J. Vu K. *The idea of "Ex Corde Ecclesiae" and Research of Vietnamese Folk Culture* Professor of Fujen Catholic University, Taiwan 1997

[Craven 1996] Craven, M. W., *Extracting Comprehensible Models from Trained Neural Networks*. PhD thesis, University of Wisconsin - Madison, 1996.

[Crawford web] Crawford, C., The extensive writings of Chris Crawford referenced in this thesis can be found on his web site at: http://www.erasmatazz.com/

[Crean 2002] Crean, B. P., O'Donoghue, D., "RADAR: Finding Analogies Using Attributes of Structure", *Proceedings of Artificial Intelligence and Cognitive Science : 13th Irish International Conference, AICS* 2002

[Culler 1975] Culler, J., *Structuralist Poetics – Structuralism, Linguistics, and the Study of Literature*, Routeledge 1975

[Dawkins 1976] Dawkins, R., *The Selfish Gene*, Oxford, UK: Oxford University Press. 1976

[Dehn 1989] Dehn, N., "Computer story-writing: The role of reconstructive and dynamic memory" (Technical Report YALE/DCS/tr712). New Haven, CT: Computer Science Department, Yale University. 1989

[Diaz-Agudo et al 2004] Diaz-Agudo, B., Gervas, P., Peinado, F., Story Plot Generation based on CBR *in Proceedings of the 7th European Conference on Case Based Reasoning. Lecture Notes in Artificial Intelligence*, 3155, 142-156, Madrid, Spain, Springer Verlag, 2004

[Dix et al 2002] Dix, J., Munoz-Avila, H., Nau, D., "Planning in a Multi-Agent Environment: Theory and Practice" *in Proceedings of AAMAS 2002* ACM 2002

[Dundes 1964] Dundes, A., *The Morphology of North American Indian Folktales*, Helsinki, Suomalainen Tiedeakatemia 1964

[Dundes 1965] Dundes, A., 1965 ,"On Computers and Folklore". *Western Folklore* 24:185-189 1965

[Dundes 1982] Dundes, A., ed. *Cinderella: A Casebook*. Madison: University of Wisconsin, 1982.

[Eagleton 1996] Eagleton, T., *Literary Theory – an Introduction*, Blackwell Publishers 1996

[Eco 1994] Eco, U., *Six Walks in the Fictional Woods* Harvard University Press 1994

[Elson 2004] Elson, D. K., Narrative Models for Summarization: Proposal for a Master's Thesis, Computer Science Sept., Columbia University 2004

[Forrest and Perlson 1994] Forrest, S., Perelson, A. S., "Self-Nonself Discrimination in a Computer" *in Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy* 1994

[Fritz web] Information on the The Fritz chess program can be found at these web links: http://www.chessbase.com/shop/product.asp?pid=189&user=&coin=
http://www.gamedevblog.com/2005/03/notes_on_fritz_.html

[Frye 1957] Frye, N.,*Anatomy of Criticism* Princeton University Press, 1957

[Gardner 1960] Gardner, M., *The Annotated Alice* Clarkson Potter 1960

[Geffner 2002] Geffner, H., "Perspectives on Artificial Intelligence Planning" in *Eighteenth national conference on Artificial intelligenc*e AAAI 2002

[Gervás et al 2004] Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: "Story Plot Generation based on CBR". 12th Conference on Applications and Innovations in Intelligent Systems, Cambridge, UK, Springer, WICS series, 2004

[Gilbert web b] Gilbert, R., interview available at: http://idlethumbs.net/display.php?id=64

[Gilbert web] Gilbert, R., 'Kalamairi Diplomacy' available at: http://www.grumpygamer.com/5279420

[Goertzel 2003] Goertzel, B., *WebMind: Self-Organizing, Network-Based Business Intelligence* Webmind 2003

[Goertzel et al 2003] Goertzel, B., Pennachin, C., Senna, A., Maia, T., Lamacie, G., "NOVAMENTE: An Integrative Architecture for Artificial General Intelligence" proceedings of

The International Joint Conference on AI in Acapulco 2003. Available online at: http://agiri.org/engine.htm

[Goethe 1983] Goethe, J. W. von, *Goethe: Essays on Art and Literature* The Collected works Vol. 3, Princeton University Press, 1983

[Gopnik and Choi 1990] Gopnik, A., Choi, S. *Do linguistic differences lead to cognitive differences?: A cross-linguistic study of semantic and cognitive development*. First Language, 10, 199-215. 1990

[Grand et al 1996] S.Grand, D.Cliff, A.Malhotra, "Creatures: Artificial Life Autonomous Software Agents for Home Entertainment", First International Conference on Autonomous Agents, 1997

[Grasbon 2001] Grasbon, D., "A Morphological Approach to Interactive Storytelling" *Cast01 Media Art Eduction* 2001

[Grasbon et al 2002] Grasbon, D., Spierling, U., Braun, N., Iurgel, I., “Setting the scene: playing digital director in interactive storytelling and creation”. *Computers & Graphics 26(1): 31-44* 2002

[Greimas 1966] Greimas, A. J., *Sémantique structurale, Recherche de méthode*. Paris: Larousse 1966

[Greimas 1990] Greimas, A. J *Narrative Semiotics and Cognitive Discourses* Pinter Publishers 1990

[Güttler 2003] Güttler, C., "Spatial Principles of Level-Design in Multi-Player First-Person Shooters" *in Proceedings of the 2nd workshop on Network and system support for games*, Redwood City, California, 158 - 170. 2003

[Hajicova et al 1996], Hajičová E., Červenka, M., Leška, O. †, Sgall, P., Ed., *Prague Linguistic Circle Papers "Travaux du cercle linguistique de Prague nouvelle série"*, Charles University, Prague 1996

[Halpin 2003] Halpin, H. R., "The plots of children and machines: The statistical and symbolic semantic analysis of narratives" (MSc. thesis). School of Informatics, University of Edinburgh 2003

[Hayes-Roth 1998] Hayes-Roth, B., "Character-Based Interactive Story Systems", *IEEE intelligent systems nov/dec* 1998

[Hobo 2004] International Hobo Inc., "Foundations of Interactive Storytelling" Based upon the presentation 'Interactive Storytelling in *Games: What it is, Why we need it, and How you do it*' presented by International Hobo at Digital Media World 2001, available online at: http://www.igda.org/writing/InteractiveStorytelling.htm

[Holland 1986] Holland, J., "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems", in *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann Publishers, San Mateo, California 1986

[IBM web] *How Deep Blue Works*, on IBM's website available at: http://www.research.ibm.com/deepblue/meet/html/d.3.2.shtml

[Jakobson 1966] Jakobson, R., *Poetics*, Polish Scientific Publishers,1966

[Johnson web] Johnson, D. M., "Lecture on Mythology" Department of Foreign Languages and Literatures SIU Carbondale, available at: http://www.siu.edu/~dfll/classics/DMJ/dmj.htm

[Jung 1953] Jung, C. G. *Synchronicity*, Princeton University Press 1953

[Jung 1934] Jung, C. G., *The Archetypes and the Collective Unconscious* Taylor & Francis Books Lt. 1934

[Koepping 1983] Koepping, K. P., *Adolf Bastian and the Psychic Unity of Mankind* University of Queensland Press 1983

[Krohling et al 2002] Krohling, R.A., Zhou, Y., Tyrrell, A. M., "Evolving FPGA-based Robot Controllers using an Evolutionary Algorithm", 2002

[Laird 1999] Van Lent, M., Laird, J. E., Buckman, J., Hartford, J., Houchard, S., Steinkraus, K. and Tedrake, R., "Intelligent Agents in Computer Games", Proceedings of the National Conference on Artificial Intelligence 1999

[Laird 2000] Laird, J. E., van Lent, M., "Human-level AI's Killer Application: Interactive Computer Games", *Proceedings of. AAAI 2000*, pp. 1171-1178. AAAI Press / The MIT Press, 2000.

[Laird 2003] Laird, J. E., "Computer Game Business", from the University of Michigan course Computer Game Design, 2003

[Lang 1996] Lang, A., *Myth, Ritual and Religion* Senate Books 1996

[Laurel 1991] Laurel, B., *Computers as Theatre*, Addison Wesley 1991

[LeBlanc 2004b] LeBlanc, M. "Mechanics, Dynamics, Aesthetics: A Formal Approach to Game Design." Lecture at Northwestern University, April 2004. Available online at: http://algorithmancy.8kindsoffun.com/MDAnwu.ppt

[Lebowitz 1983] Lebowitz, M., "Creating a story-telling universe", *in Proceedings of the 8th International Joint Conference on Artificial Intelligence* (pp. 63-65). 1983

[Levi-Strauss 1969] Levi-Strauss, C., *The raw and the cooked: Introduction to a science of mythology*, Vol. I New York: Harper and Row. 1969

[Levi-Strauss 1984] Levi-Strauss, C., Response to Propp in [Propp 1984]

[Lovelock 1987] Lovelock, J. E., *Gaia: a New Look at Life on Earth*, OUP, Oxford 1987

[Loyall 1997] Loyall, A. B., "Believable Agents: Building Interactive Personalities" Ph.D. Thesis. Technical Report CMU-CS-97-123, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. May 1997. Available online at: http://www2.cs.cmu.edu/afs/cs.cmu.edu/project/oz/web/papers.html

[Machado et al 2001] Machado, I., Paiva, A., Brna, P. "Real Characters in Virtual Stories Promoting Interactive Story-Creation Activities" ICVS 2001,LNCS 2197,127 –134. 2001

[MacNamee 2004] MacNamee, B., *Proactive Persistent Agents: Using Situational Intelligence to Create Support Characters in Character-centric computer games*, PhD. Thesis, Computer Science Sept., TCD 2004

[Magerko 2002] Magerko, B. "A Proposal for an Interactive Drama Architecture", In *Artificial Intelligence and Interactive Entertainment: Papers from the 2002 AAAI Spring Symposium*,. Menlo Park, CA. 2002

[Magerko and Laird 2003] Magerko, B. and Laird, J. "Building an Interactive Drama Architecture with a High Degree of Interactivity." in Proceedings of *1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, Darmstadt, Germany, March 2003.

[Mali 1999] Mali, A. D., "Plan Merging Plan Reuse as Satisfiablity", *ECP*, 84-96 1999

[Mandelbrot 1982] Mandelbrot, B. B., The *Fractal Geometry of Nature*, W. H. Freeman, San Francisco, 1982.

[Masahiro 1982] Masahiro, M., *The Buddha in the Robot*. Charles E. Tuttle Co., 1982

[Mateas 1997] Mateas, M., "An Oz-Centric Review of Interactive Drama and Believable Agents", Technical Report CMU-CS-97-156, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. June 1997.

[Mateas 1997] Mateas, M., "An Oz-Centric Review of Interactive Drama and Believable Agents" Technical Report CMU-CS-97-156, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. June 1997.

[Mateas 2001] . Mateas, M., "A Preliminary Poetics for Interactive Drama and Games", *Digital Creativity, 12 (3),*. 140-152. 2001 also appeared in *Proceedings of SIGGRAPH 2001*, Art Gallery, Art and Culture Papers, 51-58. 2001.

[Mateas 2001b] Mateas, M., "Expressive AI: A hybrid art and science practice", . *Leonardo: Journal of the International Society for Arts, Sciences, and Technology, 34* (2), 147-153. 2001

[Mateas 2002] Mateas, M., *Interactive Drama, Art, and Artificial Intelligence*, Ph.D. Thesis. Technical Report CMU-CS-02-206, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. December 2002

[Mateas and Sengers 1999] Mateas, M., Sengers, P., "Narrative Intelligence" in *The Working Notes of the AAAI Fall Symposium on Narrative Intelligence, Cape Cod, MA*, 1999

[Mateas and Stern 2004] Mateas, M., Stern, A., “Natural Language Understanding in Façade: Surface-text Processing” in Proceedings of Second International Conference, TIDSE 2004 2004

[Medvedev 1928] Medvedev, P. N., *The Formal Method in Literary Scholarship*, Johns Hopkins Press 1992, originally published 1928

[Meehan 1976] Meehan, J., *The metanovel: Writing stories by computer* (Technical Report YALE/DCS/tr074). New Haven, CT: Computer Science Department, Yale University 1976

[Milne 1988] Milne, P. J.,*Vladimir Propp and the Study of Structure in Biblical Hebrew Narrative* Sheffield Academic Press, 1988

[Minksy 1987] Minsky, M., *The Society of Mind*, William Heinemann Ltd., London, 1987

[Montessori 1966] Montessori, M., *The Secret of Childho*od, Fides Publishers 1966

[Morningstar and Farmer 1990] Morningstar, C., Farmer, F. R., "The Lessons of Lucasfilm's Habitat: Electric Communities", *Cyberspace: First Steps, Michael Benedikt (ed.)*, MIT Press 1990. Available online at: http://www.scara.com/~ole/literatur/LessonsOfHabitat.html

[Muñoz-Avila 2003] Muñoz-Avila, H., "On The Role of The Cases in Case-Based Planning"

[Muñoz-Avila et al 1999] Muñoz-Avila, H., McFarlane, D., Aha, D., Breslow, L., Ballas, J., Nau, D., "Using Guidelines to Constrain Interactive Case-Based HTN Planning", AIC-99-004, Naval Research Laboratory, Washington DC 1999

[Murray 1998] Murray, J. H., *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*, MIT Press 1998

[Musgrave 1995] Musgrave, F. K., "Formal Logic and Self-Expression", The George Washington University Department of Electrical Engineering and Computer Science 1995

[Musgrave and Mandelbrot 1992] Musgrave, F. K., Mandelbrot B. B, "The Art of Fractal Landscapes", *IBM Journal of Research and Development*, September 1992.

[Neal 2002] Neal, M., "An artificial immune system for continuous analysis of time-varying data" in Proceedings of 1st International Conference on Artificial Immune Systems, 2002

[Poe 1850] Poe, E. A., *The Philosophy of Composition* by Edgar Allan Poe, 1850

[Pollack et al 1994] Pollack, M. E., Joslin, D., Nunes, A., Ur, S., Ephrati, E., "Experimental Investigation of an Agent Commitment Strategy" Univ. of Pittsburgh Dept. Comp. Sci Tech Report 94-31 1994

[Pratchett et al 2002] Pratchett, T., Stewart, I., Cohen, J., *The Science of Discworld II The Globe*, Ebury Press 2002

[Propp 1968] Propp, V.. Morphology of the Folktale. University of Texas Press1968

[Propp 1984] Propp, V. Theory and History of Folklore University of Minnesota Press1984

[Rabin 2002] Rabin, S., *AI Game Programming Wisdom* Charles River Media 2002

[Rafea 1991] Rafea, A. A. El-W., "Review of Using Expert Systems in Agriculture" *Egyptian Expert Systems Interest Group Newsletter* 1991

[Reynolds 1999] Reynolds, C. W., "Steering Behaviors For Autonomous Characters", Game Developers Conference, 1999

[Riedl et al 2003] Riedl, M. O., Saretto, C. J., Young, R. M., "Managing interaction between users and agents in a multi-agent storytelling environment", 741-748 AAMAS 2003

[Rizzo et al 1999] Rizzo P., Veloso M.V., Miceli M., Cesta A., "Goal-based personalities and social behaviors in believable agents". *Applied Artificial Intelligence, 13, ,* 239-271. 1999

[Saunders and Gero 2000] Saunders, R., Gero, J. S., "Artificial Creativity: Emergent Notions of Creativity in Artificial Societies of Curious Agents", *in Proceedings of the Second International Conference on Generative Systems in the Electronic Arts*, 2000

[Saussure 1993] Saussure, F. de, *Course in general linguistics 1911*, Pergamon Press, 1993

[Schank 1982] Schank, R. C., *Dynamic Memory: A Theory of Reminding and Learning in Computers and People* Cambridge University Press 1982

[Schlovskij 1965] Schlovskij, V., "Art as Technique", *Russian Formalist Criticism*, ed. L.T. Lemon & M.J. Reis Lincoln: Univ. of Nebraska Press, 1965

[Schlovskij 1973] Schlovskij,V., "Sur la th orie de la prose" *Editions l'Age d'Homme*, Lausanne 1973

[Sgouros 1999] Sgouros, N. M., "Dynamic Generation, Management and Resolution of Interactive Plots", *Artif. Intell.* 107(1): 29-62. 1999

[Sims 1994] Sims, K., "Evolving 3D Morphology and Behavior by Competition", *Artificial Life IV Proceedings*, ed.by Brooks & Maes, MIT Press, pp.28-39., 1994

[Stork 1998] Stork, D. G. Ed., *Hal's Legacy: 2001's Computer as Dream and Reality*, MIT Press 1998

[Swartout et al 2001] Swartout, W. R., Hill, R. W., Gratch, J., Johnson, W. L., Kyriakakis, C., LaBore, C., Lindheim, R., Marsella, S., Miraglia, D., Moore, B., Morie, J., Rickel, J., Thiébaux, M., Tuch, L., Whitney, R., Douglas, J.,: "Toward the holodeck: integrating graphics, sound, character and story*" Agents 2001: 409-416* 2001

[Szilas 2002] Szilas, N., "Structural Models for Interactive Drama", First published at COSIGN-2002, September 2002, University of Augsburg, [Szilas 2003] Szilas, N., "IDtension: a Narrative Engine for Interactive Drama", *in Proceedings of TIDSE Conference* 2003

[Todorov 1981] Todorov, T., *Introduction to Poetics* The Harvester Press 1981

[Todorov 1988] Todorov, T., *Literature and its Theorists* Routledge & Kegan Paul

[Tulving, 77] Tulving, E., "Episodic and semantic memory", in *Organisation of Memory (Tulving, E. & Donaldson, W. Eds.)*, pp.381-403. Academic Press. 1977

[Turner 1992] Turner, S., *MINSTREL: A Model of Storytelling and Creativity*, Ph. D. Dissertation, University of California, Los Angeles, Technical note UCLA-AI-17-92, 1992

[Turner 1996] Turner, M. *The literary mind: The origins of thought and language*. New York: Oxford University Press 1996

[Twist 2004] Twist, J., "Simulating life, love and the Universe", an interview with Will Wright, available on bbc online at: http://news.bbc.co.uk/2/hi/technology/3645552.stm

[UEA web] http://www.uea.ac.uk/eas/admissions/courseprofiles/w800tb

[Veloso 1997] Veloso, M. M., "Merge Strategies for Multiple Case Plan Replay", ICCBR 413-424 1007

[Veselovskij 1913] Veselovskij, A., N., "Poetics" *Poetika Sjuzetov*. Saint-Petersbourg, 1913

[Watson and Marir 1994] Watson, I., Marir, F., "Case-Based Reasoning: A Review" published in *the Knowledge Engineering Review*, Vol 9 No. 4 1994

[Weyhrauch 1997] Weyhrauch, P., *Guiding Interactive Drama*. PhD Thesis, School of Computer Science, Carnegie Mellon University, Technical Report, CMU-CS-97109, Pittsburgh, PA. 1997

[Wiggins 2001] Wiggins, G. A., "Towards a more precise characterisation of creativity in AI", in Proceedings of the ICCBR 2001 Workshop on Creative Systems, 2001

[Wright 2003] Wright, W., "Dynamics For Designers" in Proceeding of the Game Developers Conference 2003

[Young 1999] Young, R. M., "Notes on the Use of Plan Structures in the Creation of Interactive Plot", in The Working Notes of the AAAI Fall Symposium on Narrative Intelligence, Cape Cod, MA, 1999

[Young 2000] Young, R. M., "Creating Interactive Narrative Structures: The Potential for AI Approaches" *in The Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, Stanford, CA, March 2000

[Ziff web] Ziff Davis' Digital Gaming in America Survey available at http://www.ziffdavis.com/

Computer Game References

The following table shows all the computer games that are referenced in this thesis, in the order in which they are mentioned in the text.

Game Title	Developer	Publisher	Year
Adventure	Will Crowther & Don Woods	N/A	1975
Zork	Marc Blank & David Lebling	Infocom	1977
SCUMM: Script Creation Utility for Maniac Mansion	Ron Gilbert & Aric Wilmunder	Lucasarts	1987
King's Quest series	Sierra	Sierra	1984-1998
Police Quest series	Sierra	Sierra	1987-1993
GTA: Vice City	Rockstar North	Take 2 Interactive	2002
Half Life	Valve Software	Vivendi Universal	1998
Mafia	Illusion Softworks	Take 2 Interactive	1998
Half Life 2	Valve Software	Vivendi Universal	2004
The Secret of Monkey Island	Lucasarts	Lucasarts	1990
Maniac Mansion: Day of The Tentacle	Lucasarts	Lucasarts	1993
Grim Fandango	Lucasarts	Lucasarts	1998
Psychonauts	Double Fine	Majesco	2005
The Legend of Zelda series	Nintendo	Nintendo	1986 - 2005

Deus Ex	Ion Storm	Eidos Interactive	2000
Thief series	Looking Glass	Eidos Interactive	1998-2000
Fable	Big Blue Box	Microsoft	2004
Everquest	Sony Online	Sony	1999
Ultima Online	Electronic Arts	Electronic Arts	1997
Habitat	Lucasarts	Lucasarts	1985
Active Worlds	Activeworlds Inc.	Activeworlds Inc.	1997
City of Heroes	Cryptic Studios	NCsoft	2004
Star Wars Galaxies	Sony Online	Lucasarts	2003
SimCity	Maxis	Maxis	1989
SimAnt	Maxis	Maxis	1991
SimEarth	Maxis	Maxis	1990
The Sims	Maxis	Maxis	2000
Ultima series	Origin Systems	Origin Systems	1981-1999
Creatures	Creature Labs	Mindscape	1997
Galapagos	Anark Studios	Electronic Arts	1997
Black & White	Lionhead Studios	Electronic Arts	2001
Unreal Tournament	Epic Games	GT Interactive	2000
Trust and Betrayal	Chris Crawford	N/A	1987
Balance of Power	Chris Crawford	N/A	1990

Appendices

Appendix A: Interpreting Propp's Corpus

Propp provided extensive background information in his appendices, and gave 44 multi-move stories mapped to his symbol notation in his Appendix III. This corresponded with descriptions of character function species that were described in the book. For example, the A villainy function has a large number of variations that are notated by A^1: abduction of a person, to A^{19}: the villain declares war. To adapt these constructions to use in the case library, a number of verbs and nouns were authored that corresponded roughly with the types of function variations found throughout the corpus. For instance, variation in the Rescue function, which has ten variations in the corpus, was reduced to the verbs 'fly', 'hide', and 'transform'. Other function types can also take the form of these same verbs. Propp provided the following table that was converted into scripts for use in OPIATE's case library:

Tale (new No.)	Move	D	E	F		A	B	C		↑	D	E	F	G		o	L	H M	J	I N		K	↓	Pr	Rs		L	Q	Ex	T		U	W*
93	I					A^{xvii}	B^3			↑	d^7	E_-^7	F_-																				
	II					α^6	B^3		F^1	↑	d^7	E_+^7	F_+^1																				
	III^1					A^{xvii}				↑								Pr^1		Rs^2				H^4	I^4								w^*
95	I					A^9	B^5				D^1	E^1	f^1										↓										
	II					$[\alpha^6]$	B_2^5	C		↑	D^1	E_-^1	$F_=$										↓										
98	I					A^9	B^5			↑	D^7	E^7	f^9										↓										
											D^1	E^1	f^1																				
	II					α^6	B_2^5	C		↑	D^7	E_-^7	F_-^9										↓										
											D^1	E_-^1	$F_=$																				
100						A^{ii}					D^3	E^3	F^{vi}					M		N													W*
101	II					A_{12}^{11}	B^7	C		↑					K^8			M		N									Ex			U	w^2
104	I			F^1		α^6	B^2	C		↑	D^1	E^1	F^1										↓									U	
													F^3							*N													
	II					$[\alpha^1]$							F^4							*N													W^*_*
																		M		N													
105	I					A^3	B^4	C	F_4^3	↑												K^7	↓										
	II	D^8	E^8	F^{82}		α^1		C		↑	D^8	E^8	F^8										↓	Pr^1	Rs^2								
106	I, II					A^1	B^4	C		↑												K^1	↓										
	III					A^1	B^4			↑	D^8	E^8											↓										
108						A^1				↑	D^8	E^8											↓	Pr^7	Rs^{10}								
																								Pr^7	Rs^1								
113						A^1		C		↑	D^1	E_-^1	F_-	G^4								K^1	↓	Pr^1	Rs^{43}								
											d^7	E^7	F^9																				
114						A^{xvi}					d^7	E^7	F_3^2											Pr	Rs^4	o		Q					W*
										↑	D^8	E^8	F^8										↓	Pr^1	Rs^2								
115	I					A^6				↑		E^7	F^2									K^5											
	II					A^{18}	B^4	C			D^1	E^1	F^1							I^6													W*

Tale (new No.)	Move	D	E	F		A	B	C		↑	D	E	F	G		o	L	H M	J	I N		K	↓	Pr	Rs		L	Q	Ex	T		U	W
125	I	D^4	E^4	F^9		A^9				↑						o																	
	II			F_7^1		A^{16}		C		↑								H^1	J^1	I^1		K^4	↓	Pr	Rs	}	L	Q	Ex			U	W*
126	I					A^3	B^2	C		↑												K^7	↓										
	II	*D^4	E^4	F^9		A^{19}	B^3	C					F^1					H^1	J^1	I^1	↓	K^3				T^2		Q					W*
127						A^6			o				F_4^3									K^3	T^2	Pr^6	Rs^6			Q	Ex			U	W*
128	I					α^1	B^2	C		↑	D^1	E^1	F^5	G^5																			w^1
	II				J^2	*A^1		C		↑			F^1	G^1								K^1											W*
131						A^1	B^1	C		↑								H^1		I^1		K^4	↓										w^0
132	I					A^5	B^1	C		↑												K_-^7	↓										
	II					[α^6]		C	F^3	↑	D^1	E^1	{ F^5 F_7^1	G^5 G^2	}					I^5			↓										
	III					*A^1		C		↑	d^7	E^7	F^9	G^1		o	L	M		N								Q	Ex			U	W*
133	I					A^1		C		↑	D^1	E_-^1	F_-																				
	II						B^4	C		↑	D^1	E^1						H^1		I^1		K^4	↓										
135						A^4		C					F^3					H^1		I^1		K^4	↓ {	Pr^4 Pr^1	Rs^7 Rs^5	}							
136	I					α^6	B^1	C			D^2	E^2										KF^2											
	II					A^1	B^2	C														KF^5											
	III					A^1		C										H^1		I^1		K^1	↓ {	Pr^4 Pr^2	Rs^7 Rs^5	}							
137	I	↑D^1	E^1	F^5		A^{19}		C										H^1		I^1			↓	Pr^4	Rs^7								
	II					α^1	B^2	C	F_2^1	↑		[K^{10}]	F_9^6				{	M M		N N		K^4	↓									U	W*
138	I					α^3	B^1	C		↑	D^2	E^2	F_4^2	G^2				H^1		I^1		K^4	↓ {	Pr^4 Pr^4	Rs^7 Rs^7								
	II					A^2		C		↑	D^1	E^2	F_9^3	G^1		. . .	. . .	III	. . .	. . .	. . .	K^4	↓										W*
	III					α^1		C		↑								M		N		K^1	↓										
139	I	↑D^9	E^9	F^9	G_5^6	A^1	B^1	C		↑								H^1		I^1		K^4									J^2		w^1↓
	II					*A^1		C		↑	D^9	E^9	F_7^1	G^1		o	L	M		N								Q_-		T^1	X		UQW*

Tale (new No.)	Move	D	E	F		A	B	C		↑	D	E	F	G		o	L	H M	J	I N		K	↓	Pr	Rs		L	Q	Ex	T		U	W*
140						A^1	B^1	C		↑	D^9	E^9		G^6	F^7					I^5		K^4	↓										W*
141	I					A^9			F^1	↑ {	D^9	E^9	F_9^6 F^9	} G^6	F^2					I^5		K^4	↓										
	II					*A^1		C		↑			F^6	G^5								K^4										U	W*
143	I					A^3		C	F^1	↑								H^1		I^1		K^7	↓										
	II					A^9				↑			F_9^6					H^1		I^1		K^1	↓										W*
144						[α^1]	M	C		↑			$F^2F_9^6$	G^1				M		N										T^3			W*
145						α^1	B^2	C		↑			F^3	G^1								K^2	↓										
148						A^1	B_2^4	C		↑								H^1		I^1		K^4	↓										W_-*
149						A^{17}	B^4	C		↑								H^2	{	I^2 I_2^6		K^4	↓										w°
150	I	D	E		X	α^6	B^2	C		↑												K^7	↓										
	II					α^5	B^2	C		↑								H^2		I^2		K^1	↓	Rs	Pr							U	w°
151						α^5	B^4	C		↑								H^2		I^2		K^1	↓										
152						A^9				↑								H^2		I^2		K^1	↓										
153						A^{18}		C		↑			F^1					H^3		I^3	X												Uw°↓X
154		↑D	E^7	$F_=^9$	G^1T^4	A^{18}		C		↑			F^1					H^3		I^3		K^4											W*
155	I-II					α^2	B^3	C		↑	D^2	E^2	F^1									K	↓										
	III	↑<Y G^2 W* F^5. . .				A^{14}	B^4	C		↑				G^2						I^{51}		K^{91}									U_-		
	IV				. . .	A^{18}	B^4	C		↑								H^1	J^1	I^1		K^1					L	Q	Ex		U	W*	}↓X
156	I					A^1	B^3	C		↑ {	D^2 D^1	E^2 E^1	F^2 F^6	G^2 G^5	}							K^4											
	II					α^4	B^4	C		↑	*D^4	E^4	F_9^6	G^5								K^2	↓									U	w^1
	III			F^2		*A^7		C		↑				G^1		o		M		N												{U	W* W* }
159	I					A^1		C		↑	. . .	. . .	. II .	. . .	. . .	. . .	. . .	. . .	II	. . .	. . .	K^4											
	II				W*	A^1		C		↑			F^9									K^4	↓	Pr^1	Rs^1								
	III					A_1^{14}	} B^4	C		↑												K^9											
								C		↑												K^4	↓	Pr^1	Rs^{12}								

Tale (new No.)	Move	D	E	F		A	B	C		↑	D	E	F	G		o	L	H M	J	I N		K	↓	Pr	Rs		L	Q	Ex	T		U	W
159	IV					a^2		C	F^1	↑	*D D^1	E E^1	F^9 F^8									KF^8	↓	Pr^1	Rs^2								
161	I					A^{19}	B^2	C		↑								H^1		I^1		K^4	↓										
	II					a^1		C		↑			F^1	G^3				H^1		I^1		KF^9											
	III					a^1		C		↑	D^9	E^9	F^4	G^6 G^5						I^5		K^4											
	IV					a^1		C		↑				G^2				H^1		I^1													W*
162		D^6	E^6	F^1		A^{11}												H^1		I^1		K^8	↓										W*
						A^1	B^3	C		↑								H^1		I^1		K^4											
163						a^1	F_6^9	C		↑				G^3																T^4			W*↓
																				I^5													W*
164	I					A^5		C		↑												K^7	↓										
	II					[a^1]	F^9	C		↑			T^4 T^4	G^3						I^5													W*↓ W *
166	I	*D^4	E^4	F^9		A^8				↑				G^1																			W*
	II					A^{10}								G^2																T^2	Q		
167	I					a^5					D	E	F^9									K^6								T		U	W*
	II				W*	A^{10}	B^3	C		↑				G^2																			
	III					a^1	B^3	C										M		N													W*↓

Figure A.1: Propp's Corpus of Tales as Series of Character Function Symbols

An example two move story script follows, adapted from example no.150 from Propp's Tables, reproduced above. Propp described this story as a list of symbols and these symbols were converted to appropriate script lines. The corresponding ISE script is included :-

Move 1	**ISE script**
D the hero is tested	Test ;
E the hero's reaction	Reaction ;
a6 an animal lack exists	Lack (neededObject ;
B2 the hero is informed	Mediation (family ;
C the hero decides on counteraction	Counteraction ;
↑ the hero leaves	Departure;
K7 the animal is caught	Liquidation (neededObject ;
↓ the hero returns	Return ;
Move 2	
a5 a lack of money exists	Lack (desiredObject ;
B2 the hero is informed	Mediation (mediator ;
C the hero decides on counteraction	Counteraction ;
↑ the hero leaves	Departure ;
H2 hero & villain compete	Struggle ;
I2 villain is defeated	Victory ;
K1 object of search is siezed by the hero	Liquidation (desiredObject ;
↓ hero returns	Return ;
Pr hero is pursued	Pursuit ;
Rs hero is rescued	Rescue ;
U villain is punished	Punishment injure (villain ;
W0 hero recieves reward	Wedding (desiredObject ;

The scripting system has both moves in the same script, but each move is represented as a seperate case in the CBP system. With no human author intervention, the CBP system evaluates which scripts to use based on the state of the game world and the story functions used so far. The presence of set roles and types of objects in a script will notify the SD as to what to look for in a story world.

Appendix B: Story Director Casting Rules

The story director bases casting decisions on the following rules. Each of the eight character roles (Villain, Donor, Helper, Mediator, Princess, King, Family, and Falsehero) have a rule that is expressed as a conditional statement, based on suitability values calculated from attitude values.

```
for(int j=1; j< 9; j++){                //go through possible roles

  switch(j){
   case 1:                              //villain
    if(!roles[0]->contains(villain)){
     villain=0;
     role* tempro;
     tempro=roles[0];
     float tempsuit=0;

     do{
//get attitude to hero
      attitude* tempat = tempro->personae->charRatings->get((int)hero);
      bool know = tempro->personae->charRatings->isThere((int)hero);
      if( tempat && tempat->rating < 0 && (tempat->rating * -1) >= tempsuit){
       tempsuit= tempat->rating * -1;
              // the following line adds the character to the list of possible
candidates for the role, along with a measure of their suitability
       roles[j]->addsuitable(tempro->personae,tempsuit);
      }
      tempro = tempro->next;
     }while(tempro->personae  != 0);
     villain=roles[j]->personae;
    }
```

```
break;

case 2:                                         //donor
 if(!roles[0]->contains(donor)){
  donor=0;
  role* tempro;
  tempro=roles[0];
  float tempsuit;

  do{
                                              //get attitude to hero
    attitude* tempat = tempro->personae->charRatings->get((int)hero);
    bool know = tempro->personae->charRatings->isThere((int)hero);
    if( !know || ( tempat && tempat->rating >= 0 ) ){
     if(tempat)tempsuit= tempat->rating*-1;
     else tempsuit=-2;
     roles[j]->addsuitable(tempro->personae,tempsuit);
    }
    tempro = tempro->next;
  }while(tempro->personae  != 0);
  donor=roles[j]->personae;
 }
break;

case 3:                                       //mediator
 if(!roles[0]->contains(mediator)){
  mediator=0;
  role* tempro;
  tempro=roles[0];

  do{
    if( tempro->personae != villain && tempro->personae != donor ){
    tempsuit= tempat->rating*-1;
    roles[j]->addsuitable(tempro->personae,tempsuit);
   }
   tempro = tempro->next;
  }while(tempro->personae  != 0);
  mediator=roles[j]->personae;
 }
break;

case 4:                                           //family
```

```
    if(!roles[0]->contains(family)){
     family=0;
     role* tempro;
     tempro=roles[0];
     float tempsuit;

     do{
                                                //get attitude to hero
attitude* tempat = tempro->personae->charRatings->get((int)hero);       if(
tempat && tempat->rating > 0  ){
        tempsuit= tempat->rating;
        roles[j]->addsuitable(tempro->personae,tempsuit);
      }
      tempro = tempro->next;
     }while(tempro->personae  != 0);
     family=roles[j]->personae;
    }
   break;

   case 5:                                        //princess
    if(!roles[0]->contains(princess)){
     princess=0;
     role* tempro;
     tempro=roles[0];

     do{
                                         //get attitude to hero & villain
attitude* tempat = tempro->personae->charRatings->get((int)hero);
 attitude* tempatvil = tempro->personae->charRatings->get((int)villain);
      float tempsuit=0;
      if( (tempat && tempat->rating > 0)  || (!tempat && tempatvil != 0 )){
       if(tempat) tempsuit= tempat->rating;
       if(tempatvil) tempsuit -= tempatvil->rating;
       roles[j]->addsuitable(tempro->personae,tempsuit);
      }
      tempro = tempro->next;
     }while(tempro->personae  != 0);
     princess=roles[j]->personae;
    }
   break;

   case 6:                                         //helper
```

```
    if(!roles[0]->contains(helper)){
     helper=0;
     role* tempro;
     tempro=roles[0];
     float tempsuit;

     do{
//get attitude to hero

attitude* tempat = tempro->personae->charRatings->get((int)hero);       if(
tempat &&  tempat->rating > 0 ){
       tempsuit= tempat->rating;
       roles[j]->addsuitable(tempro->personae,tempsuit);
      }
      tempro = tempro->next;
     }while(tempro->personae  != 0);
     helper=roles[j]->personae;
    }
   break;

   case 7:                                  //king
    if(!roles[0]->contains(king)){
     king=0;
     role* tempro;
     tempro=roles[0];
     float tempsuit;

     do{
                                       //get attitude to princess
      attitude* tempat = tempro->personae->charRatings->get((int)princess);
      bool know = tempro->personae->charRatings->isThere((int)princess);
      if( know &&  tempat && tempat->rating >= 0 ){
       tempsuit= tempat->rating;
       roles[j]->addsuitable(tempro->personae,tempsuit);
      }
      tempro = tempro->next;
     }while(tempro->personae  != 0);
     king=roles[j]->personae;
    }
   break;

   case 8:                               //false hero
```

```
    if(!roles[0]->contains(falsehero)){
     falsehero=0;
     role* tempro;
     tempro=roles[0];
     float tempsuit;

     do{
                                           //get attitude to hero & villain

attitude* tempat = tempro->personae->charRatings->get((int)hero);
 attitude* tempatvil = tempro->personae->charRatings->get((int)villain);
     if( tempat && tempat->rating < 0 && tempro->personae != villain){
       tempsuit = tempat->rating * -1;
       roles[j]->addsuitable(tempro->personae,tempsuit);
      }
      tempro = tempro->next;
     }while(tempro->personae  != 0);
     falsehero=roles[j]->personae;
    }
   break;
   default:
    break;
  }
 }
```

Appendix C: Character Modelling

This appendix shows the modelling of the character 'Trivmaj' from the third demo game, with the graphical display procedure, and the authored backstory events and behaviours. This character was authored as the initial villain for the game world, although this can change during the course of the social simulation, and characters that are socially close to this character can take the villain role as well. Figure C.1 shows the graphical character model.

Figure C.1: Trivmaj

The procedure to draw the character uses a number of simple OpenGL calls, with the animation being calculated based on trigonometric equations using the current frame, which goes incrementally from 0 to 50. Parts of the model are scaled on different 3D axes using the 'glScaled' function, which creates the effect of the robe billowing.

```
nohly::draw(){   //The character was initally named 'nohly'

 glPushMatrix();
 glTranslatef(-xposn,-yposn,0.0f);  //position
```

```
 glRotatef(-zrotn,0.0f,0.0f,1.0f);  //orientation

 bool stopped=false;
 if(speed == 0 || isWait){
  frame=0;
  stopped=true;
 }
                                  // these variables control animation
 int angle=(int)(30*sin(degtorad*(double)(frame*720/50)));
 int angle2=(int)(30*sin(degtorad*(double)(frame*360/50)));
 glRotatef(angle2,0.0f,0.0f,1.0f);

 if(!stopped) {
  if(angle == 0);
  else
      glScaled(0.9-0.1*((float)angle)/30,1.1+0.1*((float)angle)/30,1);
 }
 glTranslatef(0.0f,0.0f,0.01f);
 glColor3f(0.1,0.1,0.15);
 gluDisk(quadric,0.0, 0.31, 8, 1);
 float ang2=(float)(sin(degtorad*(double)(70)));
 float w1 = 0.25+ang2*0.08;

 for(float i=1; i<12; i++){  // draws the cloak
  float ang=(float)(sin(degtorad*(double)(70+i*360/11)));
  glColor3f(0.1+i*0.03,0.1+i*0.03,0.15+i*0.03);
  float w2 = 0.25+ang*0.08;
  gluCylinder(quadric,w1-i*0.01,w2-(i+1)*0.01,0.05,12,1);
  glTranslatef(0.0f,0.0f,0.05f);
  if(stopped)
   glRotatef(-1,1.0f,0.0f,0.0f);
  else{
   glRotatef(1+0.02*angle,1.0f,0.0f,0.0f);
   glRotatef(0.02*angle2,0.0f,1.0f,0.0f);
  }
  w1=w2;
 }

 if(!stopped){
  if(angle==0);
  else
glScaled(1.0/(0.9-0.1*((float)angle)/30),1.0/(1.1+0.1*((float)angle)/30),1);
```

```
 }

 glRotatef(-angle2,0.0f,0.0f,1.0f);

 if(stopped)
  glRotatef(5,1.0f,0.0f,0.0f);

 glTranslatef(0.0f,-0.03f,0.02f);
 glRotatef(90,1.0f,0.0f,0.0f);
 glColor3f(0.9f,0.6f,0.5f);
 glTranslatef(0.09f,0.0f,0.0f);
 gluSphere(quadric,0.05,6,6);
 glTranslatef(-0.18f,0.0f,0.0f);
 gluSphere(quadric,0.05,6,6);
 glTranslatef(0.09f,0.05f,0.0f);

 for(float k=0; k<5; k++){    // draws the nose
  glColor3f(0.82+k*0.02,0.52+k*0.02,0.42+k*0.02);
  gluCylinder(quadric,0.1-k*0.01,0.1-(k+1)*0.01,0.06,8,1);
  glTranslatef(0.0f,0.0f,0.06f);
 }
 gluSphere(quadric,0.05,8,8);
 glTranslatef(0.0f,0.0f,-0.3f);

 glColor3f(0.35f,0.35f,0.45f);   //draws the hat
 glRotatef(-110,1.0f,0.0f,0.0f);
 glTranslatef(0.0f,0.0f,0.03f);
 gluCylinder(quadric,0.3,0.0,0.1,12,1);
 glTranslatef(0.0f,0.0f,0.05f);

 for(float j=0; j<10; j++){
   glColor3f(0.35-j*0.01,0.35-j*0.01,0.4-j*0.01);
   gluCylinder(quadric,0.1-j*0.01,0.1-(j+1)*0.01,0.03,12,1);
   glTranslatef(0.0f,0.0f,0.03f);
   glRotatef(-2,1.0f,0.0f,0.0f);

  }
  frame=(frame+1)%50;   //increments the animation frame
  glPopMatrix();
}
```

The backstory events are created in the World object constructor, along with the assignment of behaviours and objects, etc. The following instructions are those relevant to the Trivmaj character.

```
...
                    // creating a magic wand object and assigning it to Trivmaj
stuff[57] = new wand();
stuff[57]->addTo(-2,-6,0);
guys[16]->addStuff((actionobject*)stuff[57]);

stuff[58] = new whirlwind((int)places[7]);        // a whirlwind object
stuff[58]->addTo(6,-6,0);
guys[16]->addStuff((actionobject*)stuff[58]);
...
   // adding behaviours to the Trivmaj character
guys[16]->addBehaviour(new whirlaway(guys[19]));
guys[16]->addBehaviour(new follow(stuff[110]));
...
   // meeting the two policemen before the game starts
guys[16]->meet(guys[7]);
guys[16]->meet(guys[8]);
...
```

The character is initialised in this way, with an easy and flexible system that allows for a lot of change in the character's state over the course of the game. This should be augmented by an author toolkit for creating and saving game world descriptors, and if the OPIATE system is to be incorporated in other game engines, further extension of the world modelling system would be necessary.

Appendix D: The OPIATE Questionnaire & Results

Questionnaire for Evaluation of the OPIATE system (OPIATE: Open-ended Proppian Interactive Adaptive Tale Engine)

After playing the demonstration game 'Bonji's Adventures in Calabria' for as long as you want, please fill out these 20 questions. The game is available for download here. It's only a couple of Megabytes.

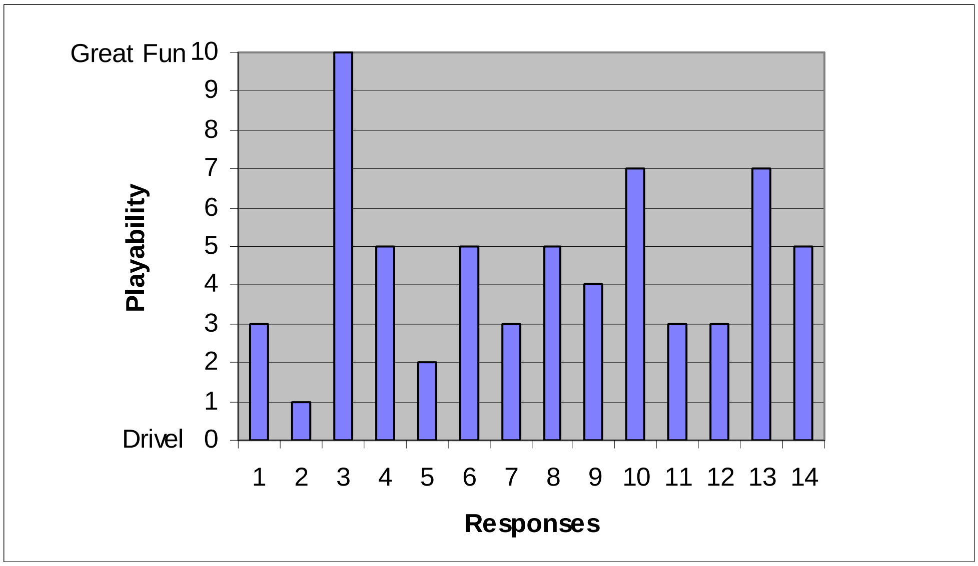
If you have played the game, and think you might want to play it again before filling this out, please do so. There is an option in the menu bar of the game to '**turn off the story AI**'. If you haven't done this yet, Please try it. It will help in answering some of the questions. Please do not fill out the questionnaire if you have not played the game. It is highly advisable to read the included 'ReadMe.txt' file before playing the game, so that you don't get stuck on anything.

Most of these questions are very subjective, so please answer the questions as you interpret them. After some questions, there is space for any extra comments, and these are most welcome. Thank you very much for taking the time to help me in my research.

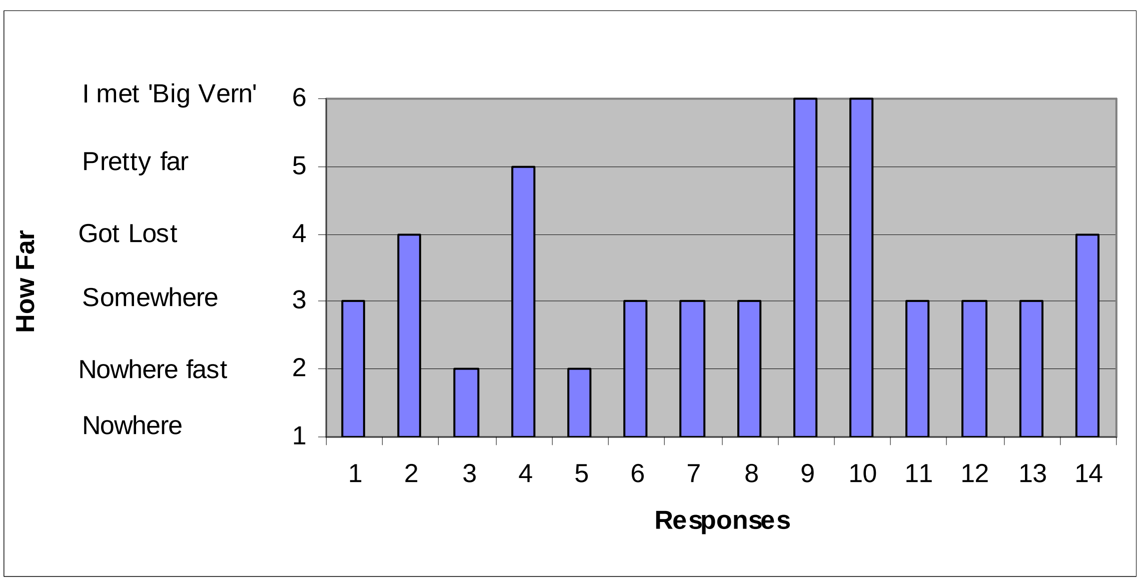
Chris

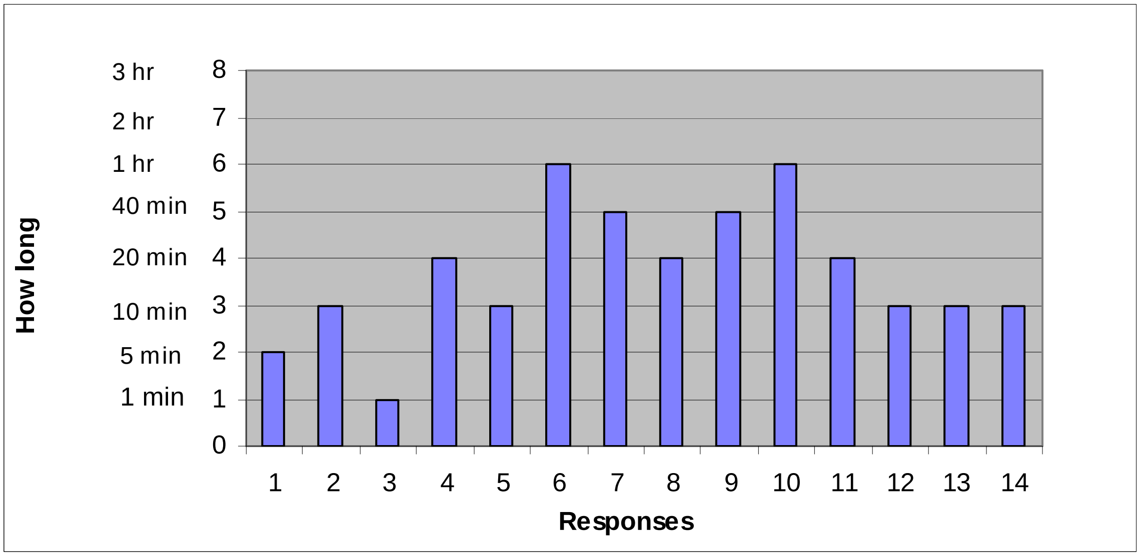
Section I

Question 1:
Did you enjoy playing this game - how 'playable' is it from 1 to 10 ?

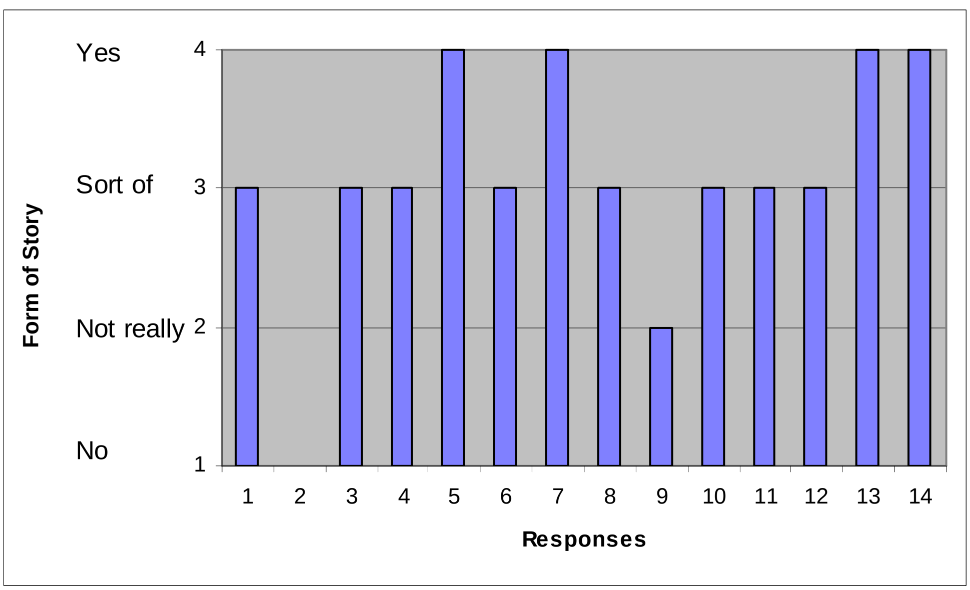


Question 2:
How far did you get in the game and how long did you spend at it?

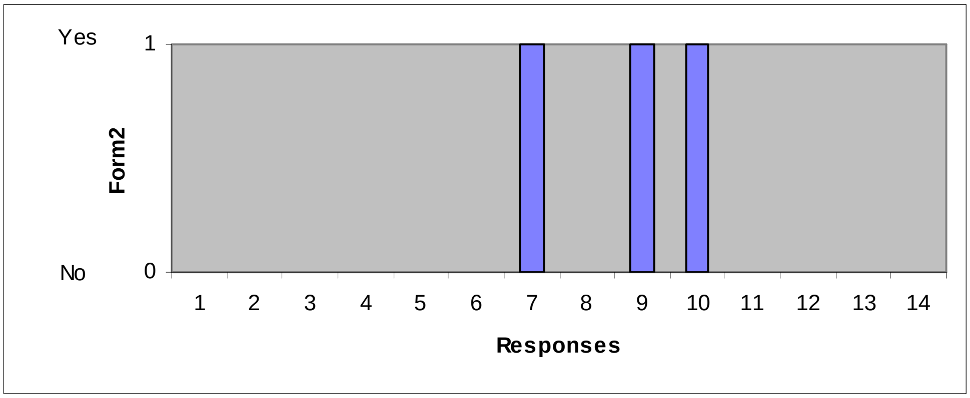




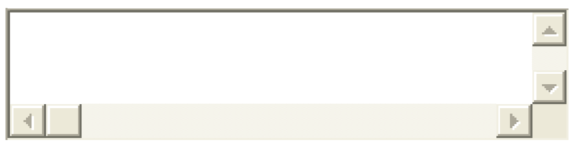
Question 3:
Did the game experience as a whole seem to **you** to be in the form of a story?



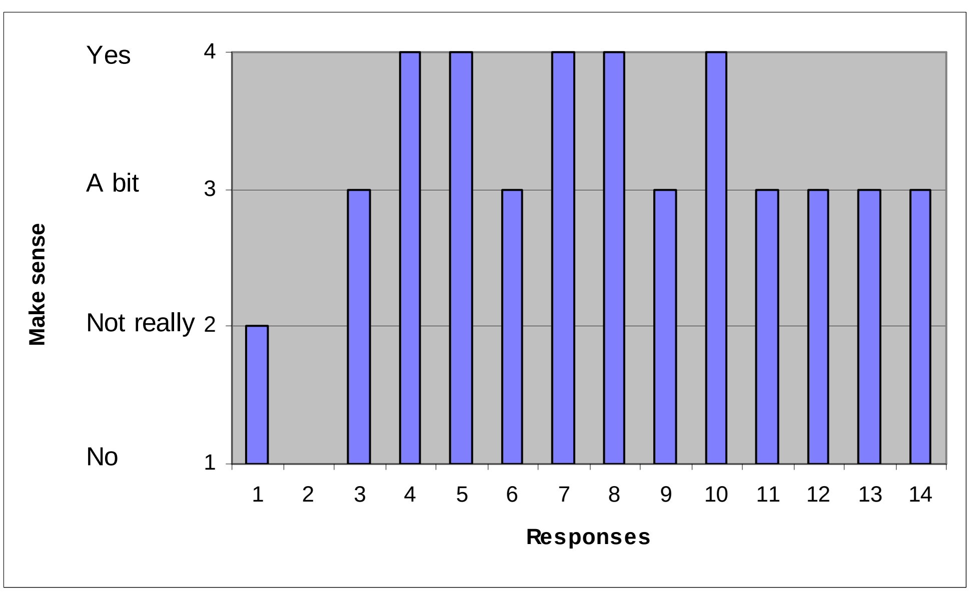
Question 4:
If you answered 'no' or 'not really' to Q.3, did **any** part of the experience seem to you to be in the form of a story?



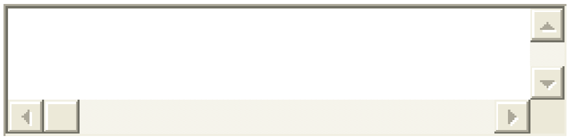
Question 5:
If you answered Yes to Q.4, what parts seemed like a story, and what parts didn't?



Question 6:
Did the series of events in the game seem to you to make sense, in the context of the cartoonish game world?

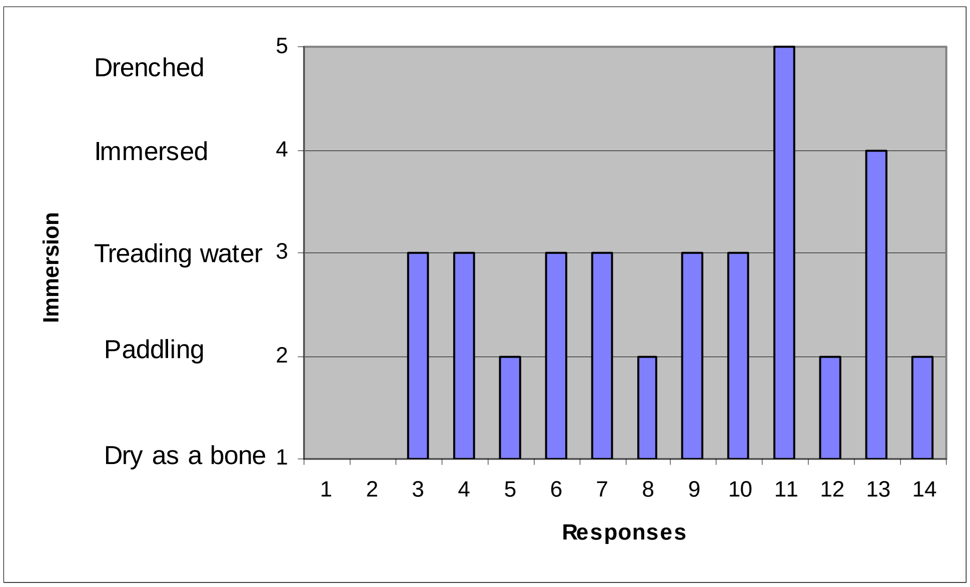


could you elaborate?

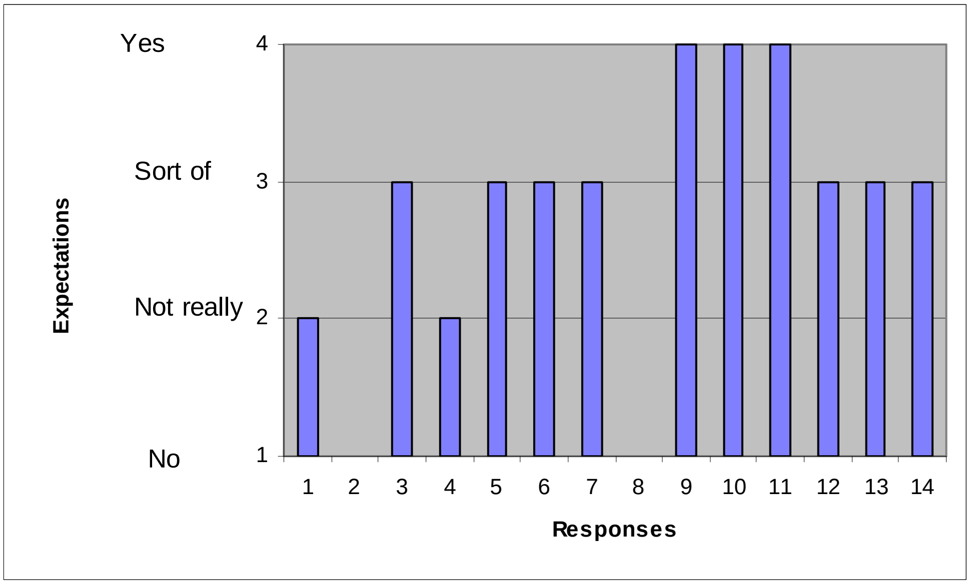


Section II

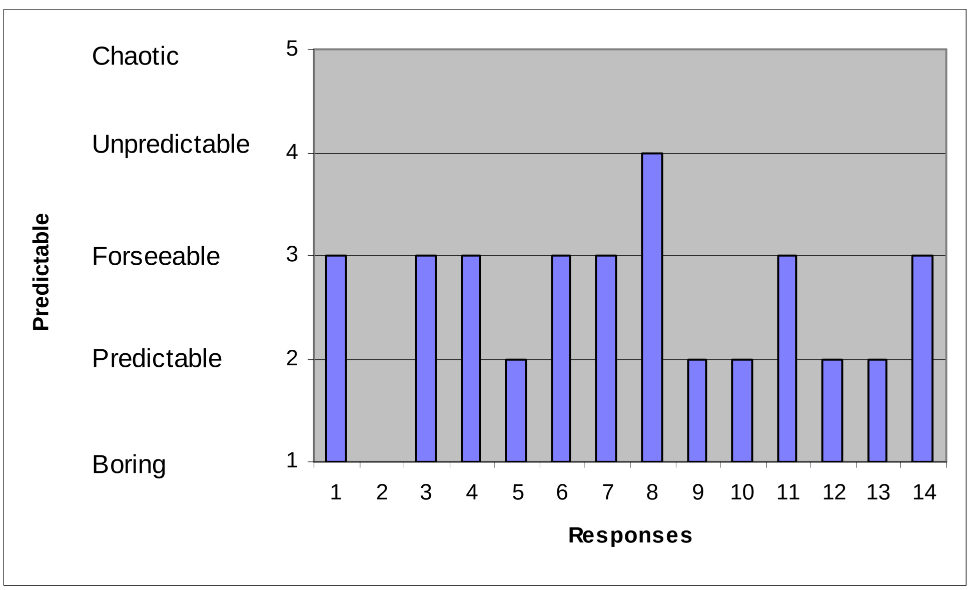
Question 7:
Did you feel immersed in the game world?

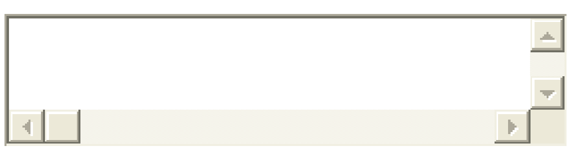


Question 8:
Did the game generate expectations in you, of what was going to happen next in the game?

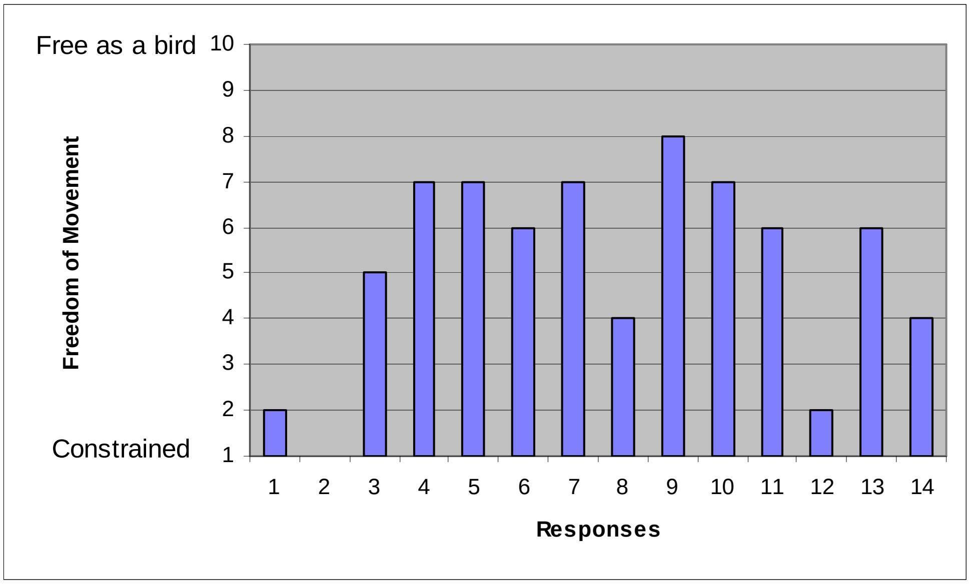


Question 9:
Did these expectations, if you had any, make the game more predictable? Please elaborate if necessary.

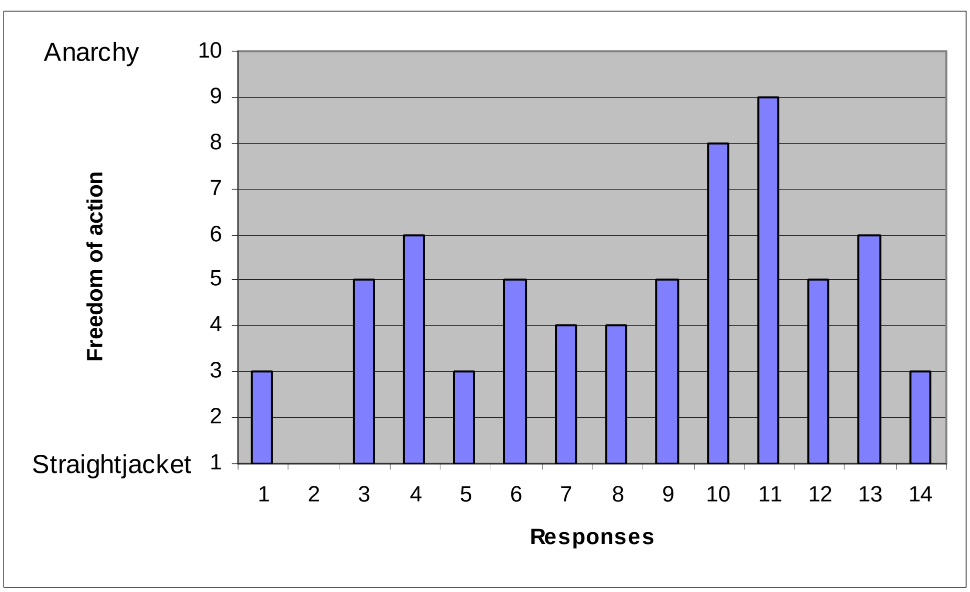




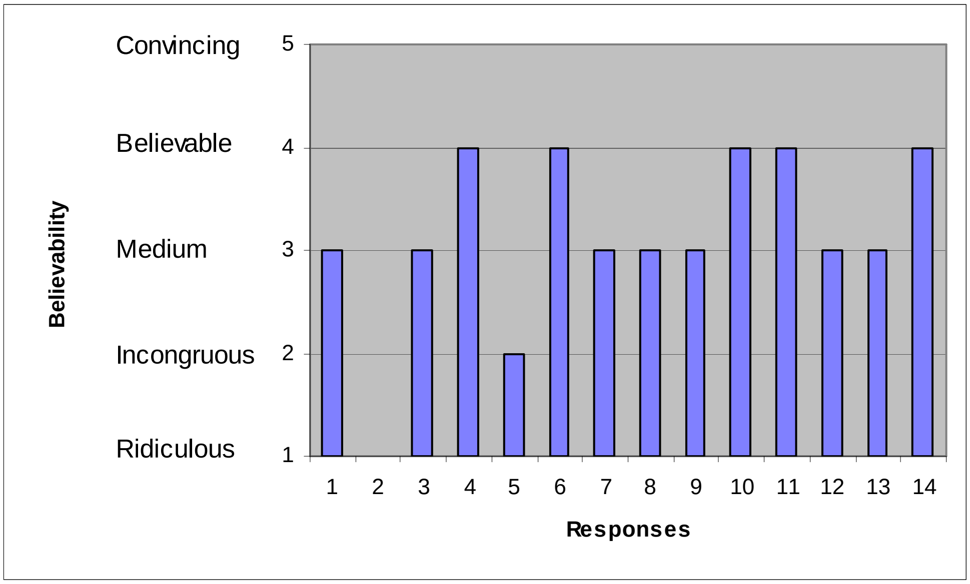
Question 10:
Please rate the degree of freedom of **movement** in the game world from 1 to 10.

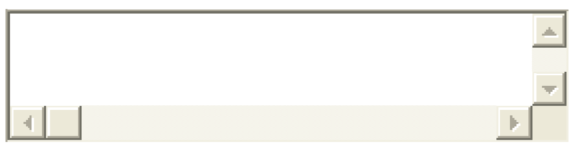


Question 11:
Please rate the degree of freedom of **action** in the game world from 1 to 10.

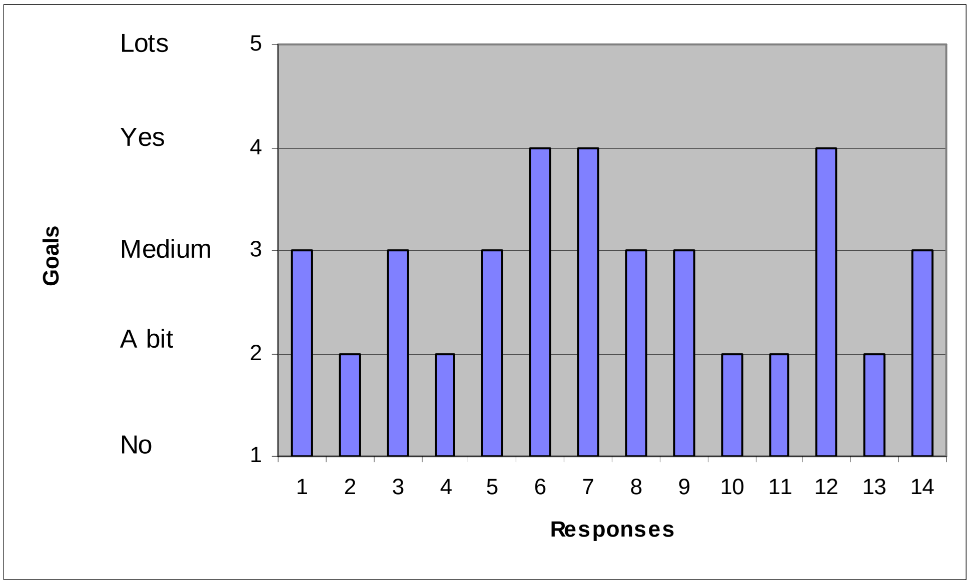


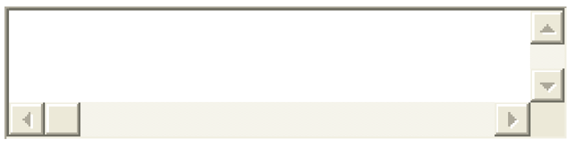
Question 12:
Did the characters seem believable? Believability in this context is not realism, but how well the characters seem to fit in their own world as distinct, independent entities. Please elaborate if possible.



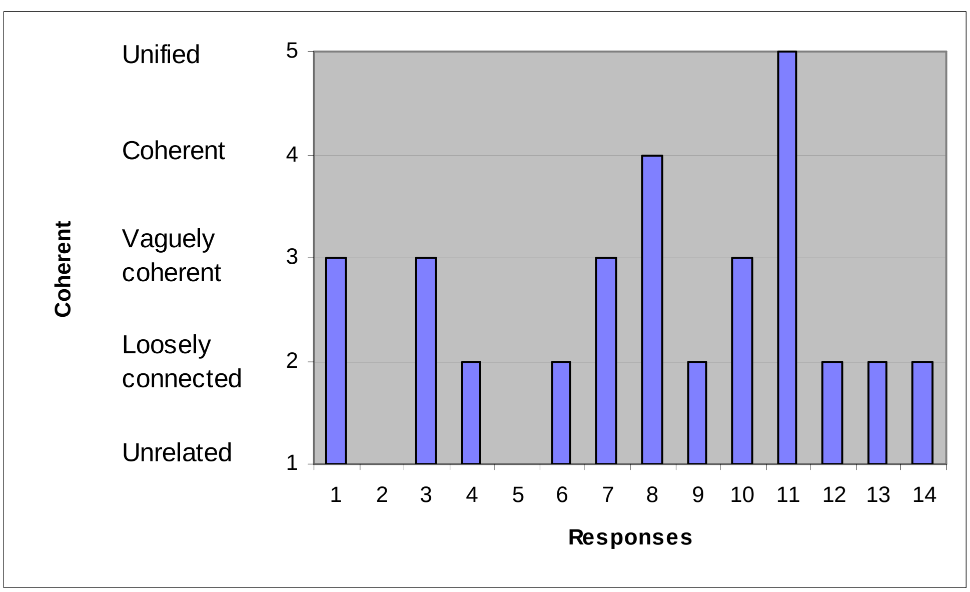


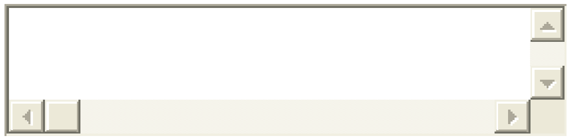
Question 13:
Did the characters seem to exhibit their own goals and attitudes? Please elaborate with examples if possible.



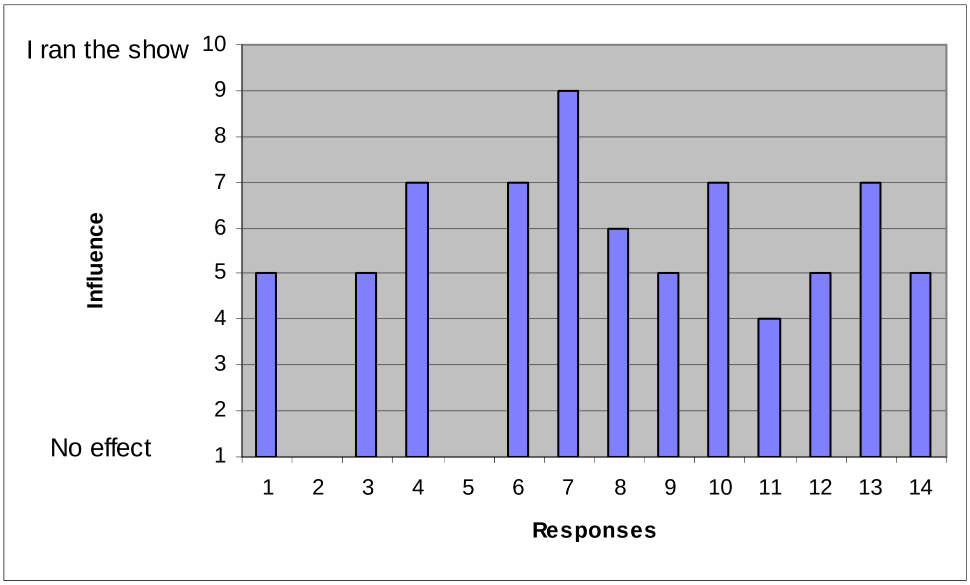


Question 14:
Did all the events seem to you to gel together as a coherent whole? - did the game seem like a series of unrelated occurences, or could you see a unity behind it?



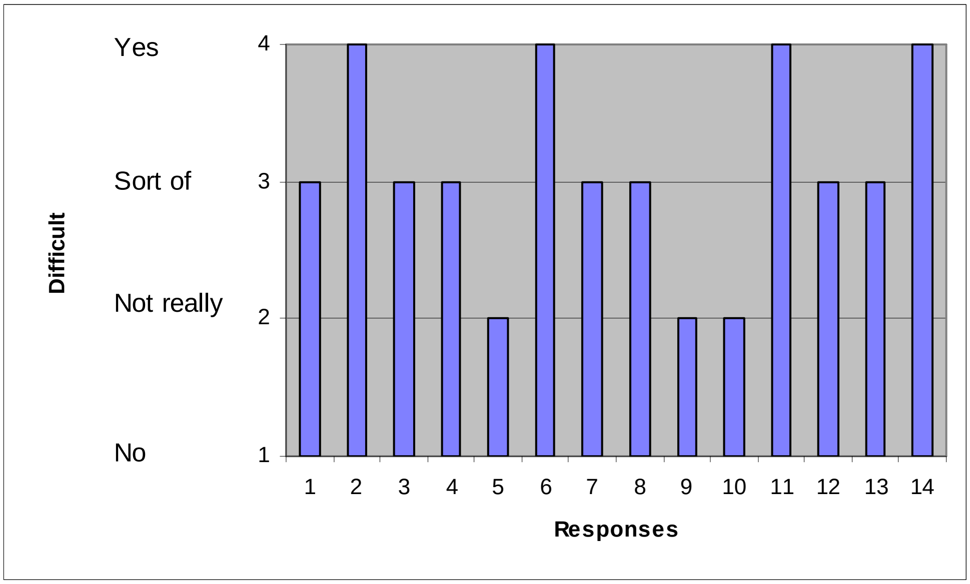


Question 15:
How much did you think the actions of your character influenced the unfolding of the story, from 1 to 10?

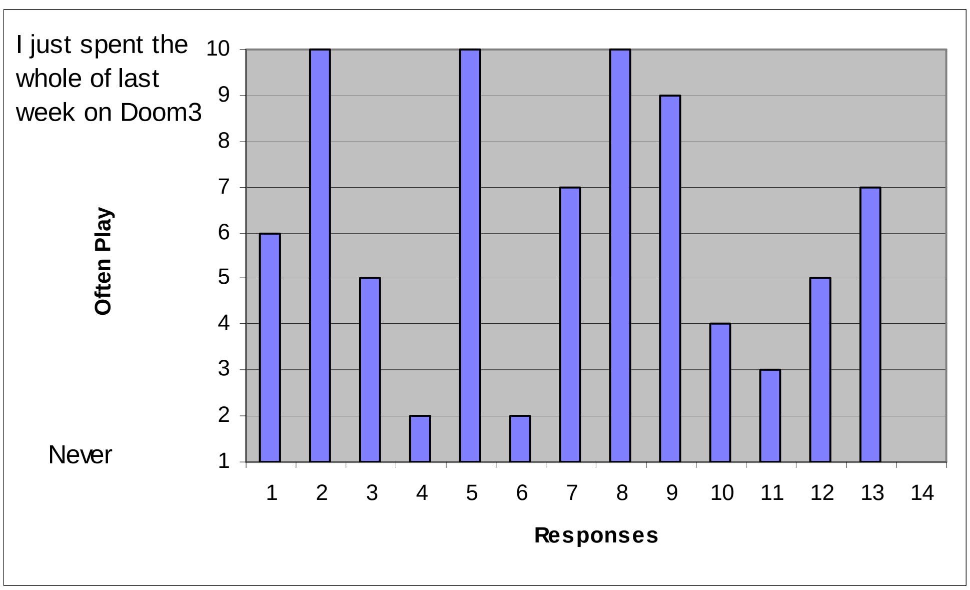


<u>Section III</u>

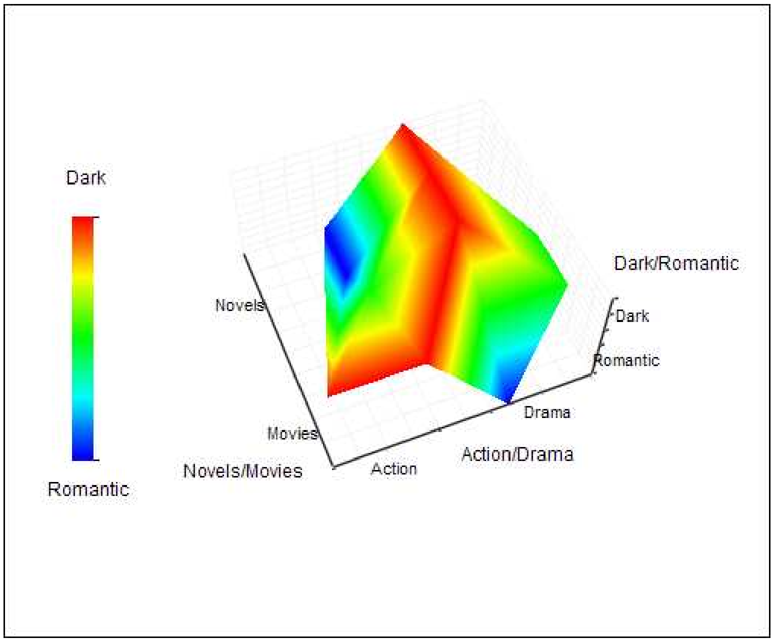
Question 16:
Did you find progress through the game difficult?



Question 17:
How often do you play computer games, from 1 to 10? (1 being never, 10 being very regularly).

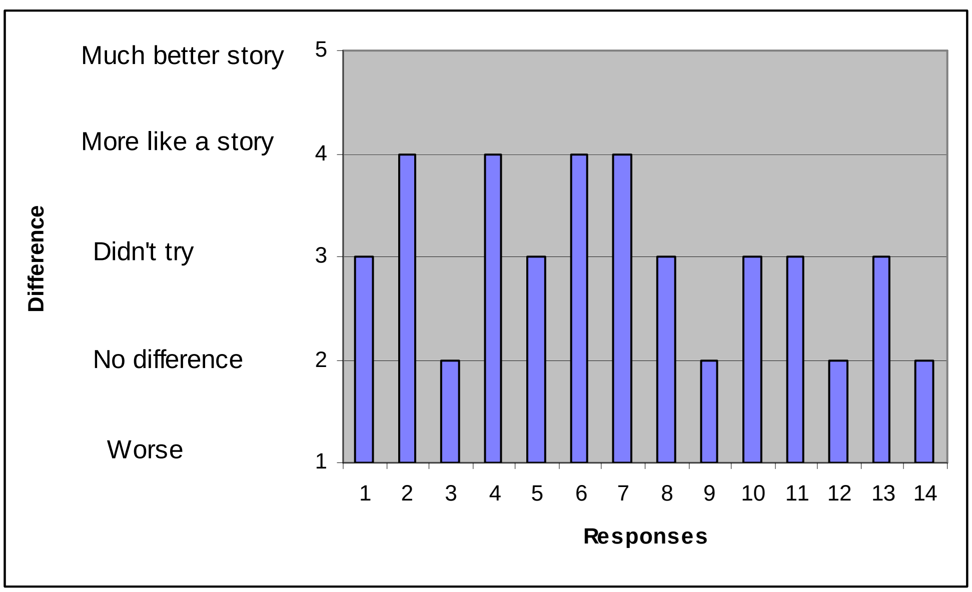


Question 18:
What sort of stories do you prefer? - action/drama, movies/novels, romantic/dark…



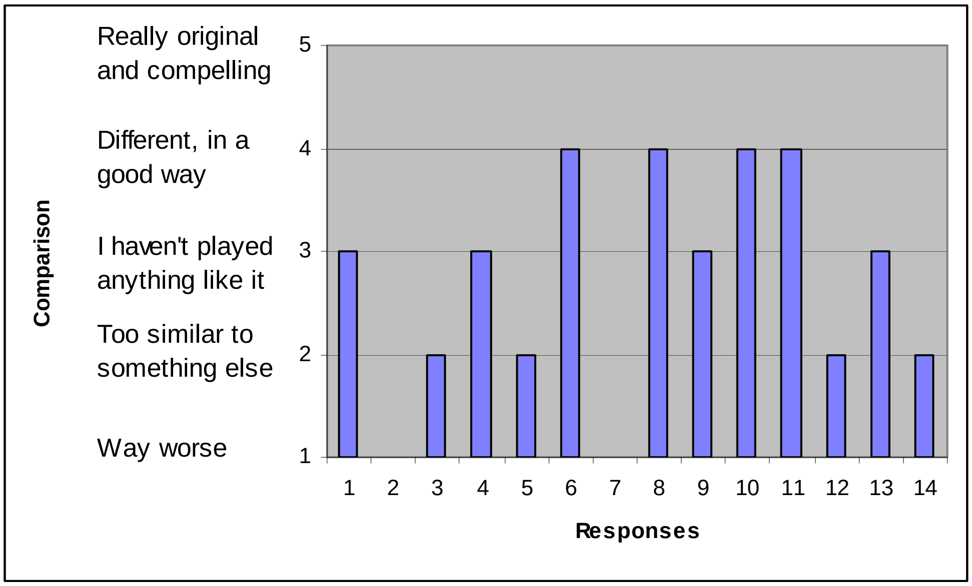
Note: This Diagram shows the 3D surface given by the points that all the respondent's answers represent, with colours clarifying the z axis.

Question 19:
Did you try the 'turn off story AI' option & did the story AI seem to make a difference to you? Please elaborate.



Note: some respondents commented that they only tried this for less than a minute, and when the SD is turned off, story events that have already been initiated happen anyway. Thus 'no difference' above.

Question 20:
How did this game compare to any other story-based computer games you may have played? (I don't mean regarding graphics quality or slick production values, but playability and originality)



Appendix E: An Example Scenario

This section will show a scenario from the execution of the OPIATE system, with Figures showing the SD planning and adaptation processes along with screenshots of the game story that the player experiences as a result of these processes.

Initial Situation

The initial situation consists of a set of characters and actionObjects situated in a set of 6 locations. Characters also have a set of attitudes that are visible to the SD. Each location's initial contents are shown in Figure E.1.

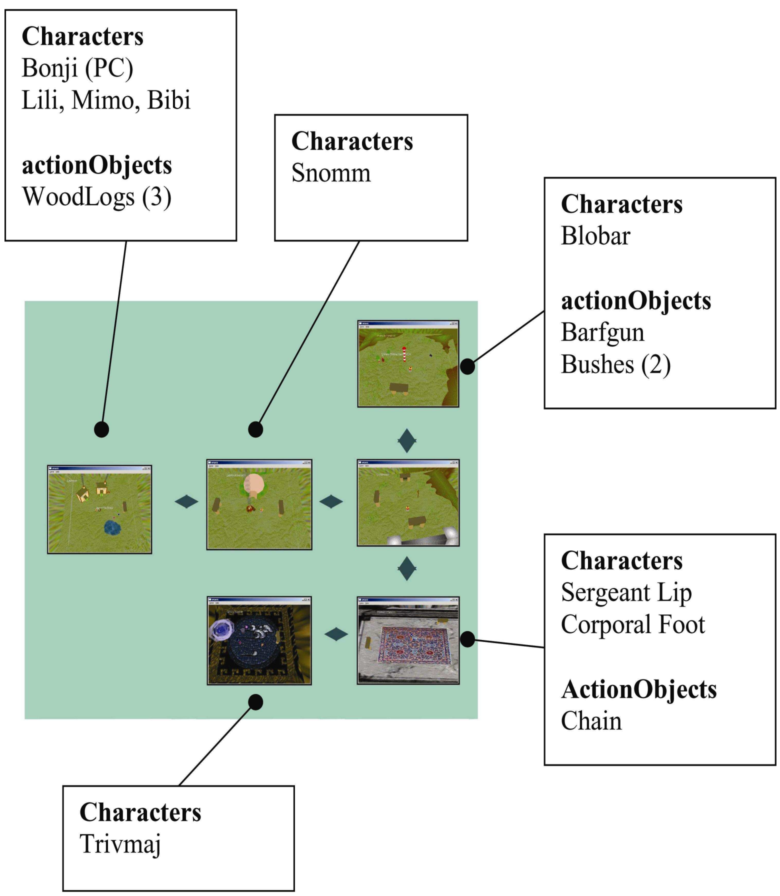


Figure E.1: Initial Location Setup and Contents. 'Bonji' is the name of the Player Character (PC). Portals between Locations are shown with dark green diamonds.

Each character has a set of initial attitudes that lie on a one-dimensional scale, from ‘like’ to ‘dislike’. The author of the game world defines these attitudes, along with the inventory of the characters. The SD agent is aware of these attitudes and casts characters into each of the eight roles based on them.

Character Name	Initial Attitudes	Initial Inventory
Lili	Likes Mimo & Bibi	
Mimo	Likes Lili & Bibi	
Bibi	Likes Lili & Mimo	
Snomm		Magic Mirror Problem
Blobar		
Sergeant Lip	Likes Trivmaj & Foot Dislikes Bonji	
Corporal Foot	Likes Trivmaj & Lip Dislikes Bonji	
Trivmaj	Dislikes Bonji	Whirlwind Wand

Figure E.2: Initial Character Setup

Each actionObject is capable of performing a different action, but these actions are classified into types of actions, defined by the attribute ‘verb’, as used in the Design chapter as a component of a character function.

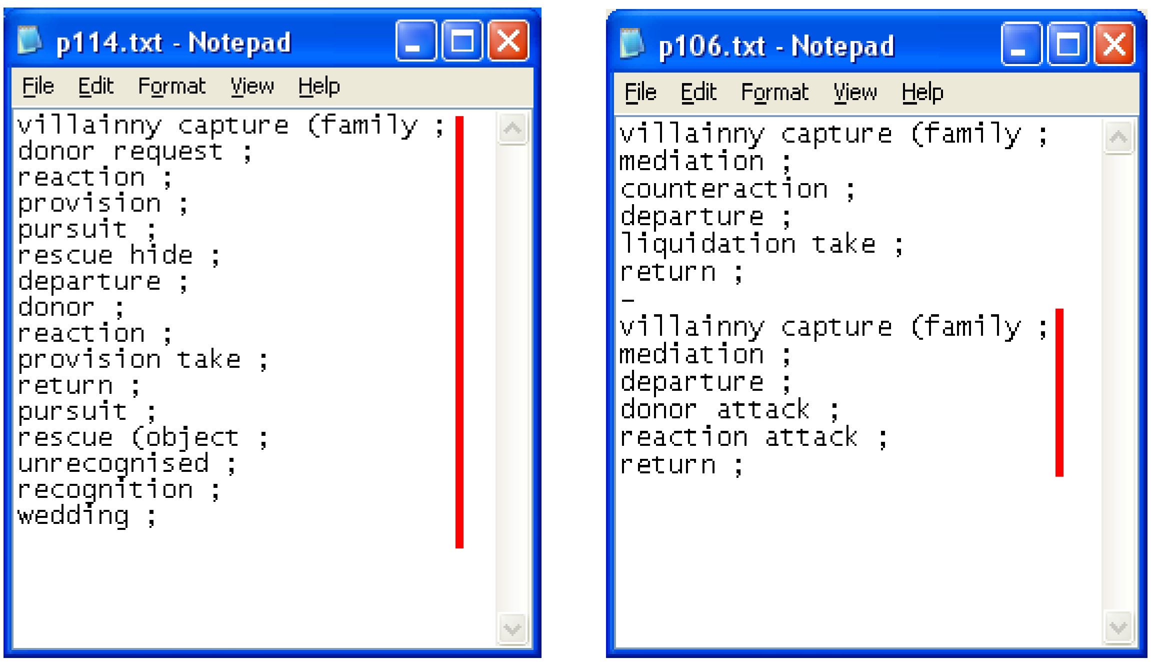
ActionObject Name	Attributes
Woodlog	Wanted
Barfgun	Verb: Attack
Bush	Verb: Hide
Chain	Verb: Capture
Magic Mirror	Verb: Attack
Problem	Verb: Request
Whirlwind	Verb: Expel
Wand	Verb: Spell

Figure E.3: ActionObject Attributes

Each actionObject action is classified by the game world author into categories given by these verbs. The SD agent is aware of these categories, and can generate sub-goals for characters to find and acquire these objects if a storyGoal requires it.

Move Selection

Move selection takes place by comparing the plot moves from the case library with the current state of the game world. Character attitudes and possessed actionObjects, and available actionObjects in the world, are used in the case selection process. Each function is assigned a suitability rating based on the heuristics given in the Design chapter. The average suitability of character functions in a move is used to rate the overall suitability of that move.



Move (a) **Move (b)**

Figure E.4: Two Moves Selected for Suitability (Marked with Red)

The suitability values of move (a) and move (b) above are calculated as shown in Figure E.5 below. These are calculated based on the equations given in the 'Story Director' section of the Design chapter.

Function #	Move (a) Suitability	Move (b) Suitability
1	3	3
2	2	1
3	1	1
4	1	2
5	3	2
6	2	1
7	1	
8	1	
9	1	
10	1	
11	1	
12	3	
13	1	
14	1	
15	1	
16	1	
Total	24	10
Average	1.5	1.6

Figure E.5: Character Function Suitability Values

These two moves are thus selected for combination because they have the highest suitability values assigned to the moves in the case library, for this particular situation.

Move Combination

Move combination is done by taking the two most suitable moves and extracting the most suitable functions from them, to create a new move that is more suitable. Character functions are transferred in *groups*, so that structural integrity of the function groups are preserved. The most suitable function from both move (a) and (b) above is the same, function number 1. However, the grouping of functions given by numbers 8,9, and 10 from move (a) have a combined value of only 2. This group is a 'donor group' of functions, concerning the provision of an actionObject to the Hero character. A donor group also exists in move (b) that has a suitability value of 4. Thus, this group will be incorporated in the new adapted move instead of the donor group from move (a).

Function #	Move (a) Suitability	Move (b) Suitability	New Move
1	3	3	3
2	2	1	2
3	1	1	1
4	1	2	1
5	3	2	3
6	2	1	2
7	1		1
8	1		2
9	1		2
10	1		1
11	1		1
12	3		3
13	1		1
14	1		1
15	1		1
16	1		1
Total	24	10	26
Average	1.5	1.6	1.625

Figure E.6: Donor Function Group Transfer for Move Combination

Move Casting

The resulting move is enacted in the game world by casting storyGoals to appropriate characters. As the game dynamics proceed, the characters' suitability ratings for each role can alter, so storyGoals are cast as they are assigned for enactment. The eight roles to be cast each have their own rules, given in the design section and appendix B. Each of the characters that are assigned a storyGoal must be first cast in a role. The series of events that are played out in the game world are described in the following table, with screenshots (Figure E.7).

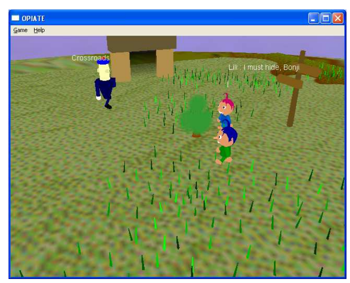
 1. The beginning – Bonji introduces himself to Lili, Mimo, and Bibi.	 2. Meanwhile, Trivmaj has the subgoal of acquiring the ‘chain’ object, for use in the ‘villainy’ storyGoal.	 3. Trivmaj carries out the ‘capture’ villainy event by kidnapping Lili.
 4. Snomm asks for help from Bonji.	 5. Bonji looks around and finds what Snomm is looking for – a woodlog for his fire.	 6. Bonji gladly gives the woodlog to Snomm.
 7. In return, Snomm gives Bonji a ‘magic mirror’ object.	 8. Sergeant Lip pursues Bonji and threatens him.	 9. Lili hides Bonji from pursuit with a ‘bush’ object.
 10. Snomm attacks Bonji.	 11. Snomm Gives Bonji back a woodlog.	 12. Bibi greets Bonji on returning.

Figure E.7: The Enactment of the Story Move shown in Figure E.6. This is one possible variation of the plot, but other characters can be cast in the storyGoals.