

Generally needs more background on institutions & norms.  
as this is the foundation of your technical  
solution.

+ Tropes

## Chapter 4 3

# Literature Review

narratology,  
interactive narrative generation.

list fields in 1st sentence.

This research covers multiple fields of study, therefore an extensive literature review covering these fields is necessary. This section starts with a look at the field of *narratology*, or narrative theory, to gain some insights into the themes and components that make up stories. Looking at different formalisms that have been created for narrative and which themes and motifs recur in stories should better inform the creation of techniques with which to generate stories.

We also examine current approaches to interactive narrative generation, focussing particularly on the use of planners, drama managers, social norms and logical modelling for describing stories. Most of these implementations are designed with multi-agent systems in mind, where the story forms a set of rules which govern these agents. The start of this part of the literature survey also examines non-interactive story generation through means of techniques such as generative grammar, in order to provide a historical context for the research that follows.

As part of the examination of implementations of interactive narrative, this review especially focuses on agent-based systems. The section concludes with an overview of emotional models that can be used to model distinct characters using agents.

## 1.1 Narratology

Narratology is a deep field with many sub-fields. This review examines the parts of it that might best inform the modelling of narrative by computers, as well as the construction of interactive narrative.

The first part of the overview of narratology examines research into categorising different types of narrative, both traditional and experimental. This draws from classic narratological texts, as well as work done on "cybertext" and experimental narrative in the interactive age. This examination of recent research into non-linear narratives is essential for the construction of interactive narratives for games or simulations.

Structuralist formalisms of narrative attempt to explain how stories work by dividing them into commonly occurring themes and motifs. This is a natural fit to the modelling of narratives by computer, especially if using an ontology. This overview of narratology starts with structuralism for this reason.

there

After the overview of structuralists' narrative models follows a section on the use of formal logic for narrative modelling. The section ends with descriptions of other types of story components, taxonomies and ontologies used in the literature.

### 1.1.1 Narrative Structure

What is the difference between a narrative and a sequence of events? If we recount the events that happened to us during the course of a day, would that "count" as a narrative? When is the retelling of events a simple listing of facts, and when is it a story?

Narratologists from Bal (2009) onwards refer to the chronological ordering of events as *fabula* (from the Russian *фабула*, "scene"), and the retelling and reordering of those events in a narrative as *syuzhet* (*сюжет*, "plot"). In fact, Russian structuralists Propp (1968) and Shklovsky (1991) were the first to use these terms in the context of narrative, before their rediscovering by modern narrative theorists.

The concepts of *fabula* and *syuzhet* are key to the understanding of narrative structure. Thanks to Aristotle's *Poetics* (Halliwell, 1986), we understand that a story must have a beginning, a middle and an end. These three parts of a story describe how the *syuzhet* are organised, which can refer to its *fabula* (chronological) events in any order. For example, the beginning of a movie could be set at the present day in the life of the protagonist, the middle could be a flashback to an earlier time in her life, and the end could return to the events following the start of the movie.

Though Aristotle identifies the beginning, middle and end as three key story divisions, other theorists divide the *syuzhet* of a story further. Joseph Campbell's work *The Hero with a Thousand Faces* (Campbell, 2008) describes how almost every story is a variation of the *Hero's Journey* (also called the *monomyth*). Like Aristotle's beginning, middle and end, the *Hero's Journey* has three acts: the Departure, the Initiation and the Return. Campbell divides these acts further into seventeen distinct stages:

#### Departure

1. The Call to Adventure
2. Refusal of the Call
3. Supernatural Aid
4. Crossing the Threshold
5. Belly of the Whale

where? in Campbell?

#### Initiation

6. The Road of Trials
7. The Meeting with the Goddess
8. Woman as Temptress
9. Atonement with the Father
10. Apotheosis
11. The Ultimate Boon

## Return

12. Refusal of the Return
13. The Magic Flight
14. Rescue from Without
15. The Crossing of the Return Threshold
16. Master of Two Worlds
17. Freedom to Live

In summary, the hero begins the story at home or in some otherwise familiar setting, where she is called away on an adventure. After possibly rejecting this call, the hero leaves home (crosses the threshold) and sets off for the land of adventure. After facing many trials and possibly defeating an enemy, the hero returns home once again. Having endured the many trials of the journey, our hero becomes stronger in spirit and character.

As a classic example, consider the plot of Tolkien's *The Hobbit*. At the start of the tale, Bilbo Baggins is a comfortable but risk-averse hobbit who refuses to leave his comfortable surroundings when Gandalf first visits ~~him~~ to send him on a quest. Of course, he does eventually leave home to fulfill the quest, returning home a changed character.

Campbell argues that the *monomyth* acts as a shared cultural memory of sorts, being the universal template for the rite-of-passage tale. He says that it is passed down as myth through many different cultures, replicating in much the same way as genes (or memes) do and that it can be thought of as a *metamyth*, or the spiritual history of humanity.

Other analysts have described several different types of commonly-seen plot. Harris (1959) makes the case for just three types of plot, having either a happy ending (where the protagonist is virtuous), an unhappy ending (with a selfish protagonist) or being a tragedy (where the protagonist is struck by fate).

Booker (2004) describes seven basic plots, where each may have either a happy or unhappy ending:

- Overcoming the monster
- Rags to riches
- The quest
- Voyage and return
- Comedy
- Tragedy
- Rebirth

Georges Polti famously divides narratives into the *thirty-six dramatic situations* (Polti, 1921), derived from his analysis of classical Greek and both classical and contemporary French texts. Polti's thirty-six types of plot are:

1. Supplication
2. Deliverance
3. Crime pursued by vengeance
4. Vengeance taken for kin upon kin
5. Pursuit

6. Disaster
7. Falling prey to cruelty/misfortune
8. Revolt
9. Daring enterprise
10. Abduction
11. The enigma
12. Obtaining
13. Enmity of kin
14. Rivalry of kin
15. Murderous adultery
16. Madness
17. Fatal imprudence
18. Involuntary crimes of love
19. Slaying of kin unrecognized
20. Self-sacrifice for an ideal
21. Self-sacrifice for kin
22. All sacrificed for passion
23. Necessity of sacrificing loved ones
24. Rivalry of superior vs. inferior
25. Adultery
26. Crimes of love
27. Discovery of the dishonour of a loved one
28. Obstacles to love
29. An enemy loved
30. Ambition
31. Conflict with a god
32. Mistaken jealousy
33. Erroneous judgment
34. Remorse
35. Recovery of a lost one
36. Loss of loved ones

Can you put these  
in two columns?

Author Kurt Vonnegut describes several different "shapes" of stories in his rejected Master's thesis Vonnegut (2009). He plots the events of a story as points in a 2D space with "Beginning-End" (time) on the X-axis and "Ill Fortune-Great Fortune" (fortune of protagonist) on the Y-axis. Figure 1-1 shows two examples of story shapes. In the "Man in Hole" example, things start out well for the story's protagonist, but then some misfortune falls on them. They spend much of the story overcoming challenges before finally ending the story with a triumph. In the "Old Testament" example, the protagonist(s) experience gradually increasing levels of fortune, before finally being "cast down" to a grave level of misfortune at the end of the story.

All of the above categorisations describe stories as a whole, sorting narratives into one of several different "types" of story. This works well for linear narratives within traditional media,

Vonnegut's visualisation seems a bit different to the above. Can you summarise the above before moving on. What do they have in common? then being different?

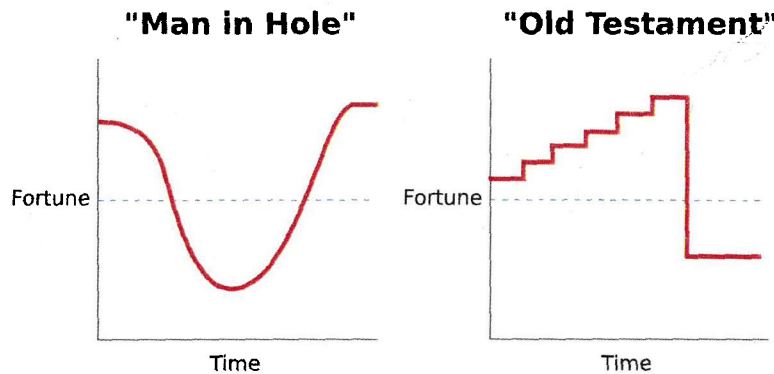


Figure 1-1: Two of Kurt Vonnegut's "Shapes of Stories"

but what about experimental and non-linear narratives? In the next section, we review the literature relating to these less traditional types of stories.

### 1.1.2 Types of Narrative

The rise of the Web in the 1990s brought with it great interest in the future of narrative in cyberspace. Aarseth's work, *Cybertext* (Aarseth, 1997) describes the creation of a new form of narrative, for which he coins the term *ergodic literature* (from the Greek words *ergon* and *hodos*, meaning 'work' and 'path'). In this new form of narrative, some amount of work or effort is required by the reader in order to traverse the path that the story takes.

Aarseth makes a distinction between the narrative as written by the author, and the way in which it is traversed by the reader, calling the former *textons* and the latter *scriptons*. In ergodic literature, the *scripton* is produced by the effort that the reader goes through in interpreting the *texton*. In the context of a game, it is as though the game interface is a gateway that allows access to the narrative at different times. Using classical music as a metaphor, the *texton* can be thought of as the *score*, and the *scripton* the *performance*.

In section ??, we assert that how generative a narrative is and its level of interactivity are two different variables in an experimental narrative. However, Aarseth identifies seven different methods of story traversal: *dynamics*, *determinability*, *transiency*, *perspective*, *access*, *linking* and *user function*.

**Dynamics** describe whether or not the content and number of *scriptons* changes. In a simple, static story with branching choices (such as in a *Choose your own adventure* story), both the number of *textons* and *scriptons* are fixed, since all paths have been written out beforehand. A dynamic story would still have a fixed number of *textons*, but the *scriptons* would be generated as the user traverses the path of the narrative.

**Determinability** is how deterministic the narrative is, whether or not the same interactions will result in the same *scripton* being produced.

**Transiency** means to what extent *scriptons* are produced as time flows, or whether user interactions are required to produce them.

**Perspective** is whether or not the user/reader plays a role as a character in the narrative.

← What's your emph path? emph all uses or first use? be consistent.

Be ambient with use of color in definition list.

**Access:** if a user has access to all scriptons at any point in traversing the narrative, or whether their access is restricted.

**Linking** means whether or not parts of the scripton are linked to other parts, and whether these links are conditional (if they rely on a user having already traversed part of the scripton).

**User functions:** the functions the user uses to traverse the text. This could be interpretive (which is implicit in any traversal of the text), explorative (traversing the scripton according to whim) or configurative (specifying parts of the scripton in advance), for example.

By performing correspondence analysis (a process similar to principle component analysis) on a diverse corpus of 23 texts "*ranging from ancient China to the Internet*", Aarseth filters these seven variables down into two numerical axes which account for 49 percent of the variation between stories. Using these axes, he groups classic tales such as *Moby Dick* and more experimental narratives such as William Gibson's *Agrippa* and Michael Joyce's *Afternoon*. By grouping these stories into categories, he intends to show how emerging media are enabling new types of story.

Chris Crawford's *Chris Crawford on Interactive Storytelling* (Crawford, 2012) provides a scathing assessment of the relationship between narrative theory and computer science. A veteran of the games industry, he argues that 'soft' science theories such as those of Aarseth et al are entirely removed from 'hard' science, and are therefore an example of bubble intellectualism and impossible to implement.

Crawford himself provides a useful examination of experimental narrative in computer games, defining interactivity as:

A cyclic process between two or more active agents in which each agent alternately listens, thinks, and speaks.

He argues that for game narratives to be truly interactive, they must be more social. Characters in a story must be able to react with the player as though they were people in real life. In turn, the player should have some degree of freedom in the way in which they interact. Rather than presenting branching story points as choices, a better way to interact would be socially, through talking to agents in the game. This is the approach that *Faade* takes (Mateas and Stern, 2003), which Crawford acknowledges as the most successful attempt at interactive storytelling to date. A detailed description of *Faade*'s implementation appears in section ??.

In order to determine whether Crawford's assertion that narratology research is too far removed from its practical implementations to be of use, we provide an overview of these implementations and their underlying research in section 1.2. Has narrative theory research informed the creation of computer-generated or interactive narrative at all, or do they all take approaches grounded in computer science and artificial intelligence? If narrative theory has not been used, then we must ask another question: "why not?"

X



### 1.1.3 Structuralist Formalisms of Narrative

Attempts to organise recurring themes, roles and motifs of narrative go back at least a century. The Aarne-Thompson tale-type index (Aarne and Thompson, 1987), first published in 1910 and later refined by Stith Thompson in 1928 and 1961, is well known amongst folklorists as a classification and analysis method for traditional folktales and myths. Aarne-Thompson's index is a taxonomy of tale themes, arranging tales into categories such as *animal tales* and *jokes and anecdotes*, and then sub-categories (*tales of magic* and *numskull [sic] stories* being two examples). This taxonomy is only two levels deep however, and only serves as a useful way to categorise individual stories or tales. In order to break down and analyse components of tales, we must dig deeper.

In *Structural Anthropology*, Claude Lévi-Strauss seeks to discover why myths and legends are so similar across cultures and history (Lévi-Strauss, 2008). He concludes that there are global laws that govern the way in which people create stories, therefore these laws can be modelled as a set of rules for describing myths.

His theory is that myths describe opposing forces which are resolved through mediation. The example he gives in *Structural Anthropology* describes how Native American legends often contain 'trickster' characters in the form of ravens or coyotes. As scavenging animals, these tricksters symbolically act as mediators between life and death.

Like much of early narrative theory, there is no rigorous evaluation of Lévi-Strauss' ideas, leaving them feeling a little too opinionated and arbitrary. While interesting, Lévi-Strauss' ideas bring us no closer to developing a formal model of narrative structure. For that, we must go even further back in time, and turn to Vladimir Propp.

#### Propp's Morphology of the Folktale

A notable narrative structuralist is Vladimir Propp, creator of *The Morphology of the Folktale* (Propp (1968), a formalism for Russian folktales. Propp's formalism, though originally limited in scope, generalises well, and is still used by researchers to procedurally generate stories Grasbon and Braun (2001); Gervás et al. (2005); Hartmann et al. (2005). Drawing from a corpus of one hundred Russian folktales, Propp identifies thirty-one distinct *story functions*, each of which is identified with a number and symbol. These functions are executed by characters following certain roles, each of which has a *sphere of action* consisting of the functions that they are able to perform at any given point of the story. Stories are created by chaining story functions together, with subplots expressed as parallel chains of story functions.

In this formalism, characters have *roles*, such as *hero*, *villain*, *dispatcher*, *false hero*, and more. Characters performing a certain role are able to perform a subset of *story functions*, which are actions that make the narrative progress. For example, the *dispatcher* might send the *hero* on a quest, or the *victim* may issue an *interdiction* to the *villain*, which is then violated.

Propp defines a total of 31 distinct story functions, each of which is given a number and symbol in order to create a succinct way of describing entire stories. Examples of such functions

find a 1952 reference.

A bit too colloquial. seeming?

by? filling?

Why are these cited (textual)?

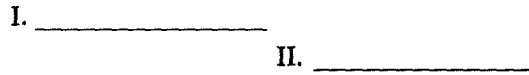


Figure 1-2: One Propp function following another

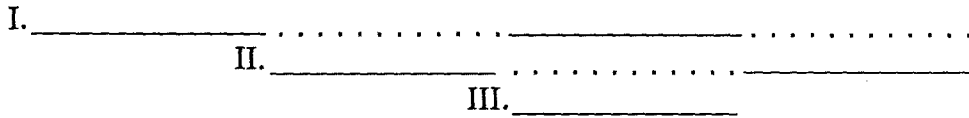


Figure 1-3: Multiple simultaneous functions

are:

- One of the members of a family absents himself from home: *absentation*.
- An interdiction is addressed to the hero: *interdiction*.
- The victim submits to deception and thereby unwittingly helps his enemy: *complicity*.
- The villain causes harm or injury to a member of the family: *villainy*.

Each of these functions can vary to some degree. For example, the *villainy* function can be realised as one of 19 distinct forms of villainous deed, including *the villain abducts a person*, *the villain seizes the daylight*, and *the villain makes a threat of cannibalism*.

These functions are enacted by characters following certain roles. Each role (or *dramatis personae* in Propp's definition) has a sphere of action consisting of the functions that they are able to perform at any point in the story. Propp defines seven roles that have distinct spheres of action: *villain*, *donor*, *helper*, *princess*, *dispatcher*, *hero*, and *false hero*.

← duplication

Though Propp defines each *dramatis personae* as being distinct (characters can only play one role at a time), it is simple to extend the idea to allow for overlapping roles. For example, a victim could also be a donor, so the set of functions they can perform would be the union of both the victim and donors' permitted function sets. Propp does not explore this possibility in *The Morphology of the Folktale*, however.

Not sure the lit survey is the place for talking about extensions.

In a typical story, one story function will follow another as the tale progresses in a sequential series of cause and effect (figure 1-2). However, Propp's formalism also allows for simultaneous story functions to be occurring at once (figure 1-3).

Though flexible, Propp's formalism is limited in its expressiveness. All story functions describe events at the same level of abstraction, describing one event after another. Also, Propp insists that the story functions occur in a prescribed order. Later French structuralists such as ~~Bremond~~ and ~~Cancalon~~ (1980), ~~Greimas~~ (1983) and ~~Todorov~~ (1969) address the latter problem by generalising Propp's work outside of Russian Folktales, though each represents only incremental improvements on Propp, lacking a means of nesting story functions to create abstractions. (Barthes and Duisit, 1975) broadly describe hierarchically composing *narrative units* but, lacking implementation details, these can only be used as a template from which to build a new narrative model.

Needs citation (textual).



1. Attempt  $\rightarrow$  Plan + Application  
 $\Rightarrow$  MOTIVATE(Plan, Application)
2. Application  $\rightarrow$  (Preaction)\* + Action + Consequence  
 $\Rightarrow$  Allow(AND(Preaction, Preaction, ...),  
 {CAUSE | INITIATE | ALLOW}(Action, Consequence))

Figure 1-4: Example rules from Rumelhart's story grammar

### 1.1.4 Other Types of "Story Component"

Lehnert's *plot units* are a more recent narrative formalism Lehnert (1981). However, these plot units only describe stories as three types of event: positive, negative and mental. These events occur with respect to a single character in the story, so an author must always author story components with concrete characters in mind, making them difficult to re-use. Similarly to Propp's system, the order of composition must always be in a certain sequence, and plot units cannot refer to other plot units. Again, we are left without a means of creating abstractions for our story components. In the "TropICAL: a DSL for Tropes" section, we describe how tropes allow the nesting of components to allow story authors to create their own abstractions, addressing this issue.

Similar

address this issue by describing

move to the head

## 1.2 Implementations of Experimental Narrative

### 1.2.1 Story Generation

Inspired by Chomsky's theories of generative grammar (Chomsky and Halle, 1968), researchers in generative narrative strive to build their own 'universal grammar' for narrative.

provides

Rumelhart (1975) is one early and influential model for the grammatical generation of natural language. Figure 1-4 shows two example rewrite rules from this grammar. The '+' symbol denotes items happening in a sequence, the '|' showing possible alternatives. '\*' denotes one or more item being generated. Capitalised words (such as ALLOW, MOTIVATE, etc) describe relationships between items. For example, MOTIVATE is a relationship between a character's thought and their reaction to that thought.

hmm.  
if you are going to describe some of the syntax then define it all.  
What about  $\rightarrow \Rightarrow$ ?

This generative grammar approach can be seen in systems such as Pemberton's GESTER (Pemberton, 1989), which generates stories based on a grammar synthesised from old French epic tales. Lang (1999) describes a declarative model for narrative, consisting of lists of first-order predicate calculus expressions. These expressions describe events, states, goals and beliefs which combine to form a narrative. More specifically, it combines:

- A **grammar interpreter** to search for a sequence of grammar rewrites which would produce a convincing narrative.
- A set of **temporal predicates** to describe the occurrence of events over time and enforce temporal constraints on story components.
- A **world model** which describes the set of actions that characters may perform and fluents that may alter over the course of the narrative.

X grammar

(This was only inspired by generative grammar right?)

Could we a textual writer?

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

GEORGE WAS VERY THIRSTY. GEORGE WANTED TO GET NEAR SOME WATER. GEORGE WALKED FROM HIS PATCH OF GROUND ACROSS THE MEADOW THROUGH THE VALLEY TO A RIVER. GEORGE FELL INTO THE WATER. GEORGE WANTED TO GET NEAR THE VALLEY. GEORGE COULDN'T GET NEAR THE VALLEY. GEORGE WANTED TO GET NEAR THE MEADOW. GEORGE COULDN'T GET NEAR THE MEADOW. WILMA WANTED TO GET NEAR GEORGE. WILMA GRABBED GEORGE WITH HER CLAW. WILMA TOOK GEORGE FROM THE RIVER THROUGH THE VALLEY TO THE MEADOW. GEORGE WAS DEVOTED TO WILMA. GEORGE OWED EVERYTHING TO WILMA. WILMA LET GO OF GEORGE. GEORGE FELL TO THE MEADOW. THE END.

Figure 1-5: Example TALE-SPIN output

- A **natural language output unit**, which takes the sequence of events produced by the story grammar and converts it into readable natural language sentences.

This combination of using a grammar interpreter, world model and natural language output unit is especially common amongst generative grammar approaches.

While generative grammar approaches may be effective for procedurally creating prose, they are less well suited to the creation of *interactive* narratives. Once the grammar rewrite rules are specified, the user is entirely passive, unable to affect the way in which the story is being generated. For this to happen, the narrative needs to be part of a system that reacts to the actions of the user, such as <sup>in</sup> a computer game.

James Meehan's TALE-SPIN (Meehan, 1977) is an influential early approach to story generation using planning. In TALE-SPIN, the author describes a story domain, its characters and their goals, and a natural language story is produced as output. It works by using a problem-solver to resolve each character's goals over the story domain. Figure 1-5 is an example of TALE-SPIN's output.

TALE-SPIN's strong planning component is evident in the reading of its output. Sentences are terse, with one event leading directly to another in order to achieve some goal. One problem with this character-led approach is that the goals of the author are not necessarily taken into account. If the author intends for a character to die at some point in the story, it seems unnatural for a character to have the goal of dying to fulfill this intention.

Turner criticises TALE-SPIN's stories as seeming "pointless and somewhat boring" (Turner and Dyer, 1986), going on to create the MINSTREL system for story generation (Turner, 1993). Using the legendary world of King Arthur's court as a story domain, MINSTREL strives to

generate stories with an authorial purpose.

MINSTREL attempts to address TALE-SPIN's shortcomings by introducing two types of schema: author-schemas and character-schemas, both of which combine to represent the elements of a story. The author-schemas describe the goals of the author of the system, allowing story creators more control over the structure and content of their narrative. This allows authors to specify a 'point' or moral to their story, something that is not possible to achieve with TALE-SPIN. Character-schemas describe character-level goals in a similar manner to those of TALE-SPIN's.

The comparison of TALE-SPIN and MINSTREL highlights a challenge that has dominated story generation for decades: the balance of *character* and *plot*. Especially with approaches based on multi-agent systems, the regulation of character actions to conform to an underlying theme or structure is a challenging problem.

\* Probably needs a ref.

However, modelling characters with agents is a promising approach to take in order to gain a story which is both generative and interactive. In such a system, an author can specify the story world and character models, creating the 'big picture', and the agents would be able to fill in the details (such as dialogue, sub-plots, and relationships).

achieve ?

same paragraph? →

An apt metaphor would be that of animation. In large animation studios such as Disney, the lead animator draws the key frames of a sequence, and a team of other animators work to fill in the gaps in between. This is what the combination of a managed narrative with agents could achieve: the author would be the 'lead animator' in such a system, with the agents being the assistants.

source - Any ref?

So far into this literature review, we have some idea of where a promising approach may lie. However, the implementations until now have still focused mainly on *generation*, and little on *interaction*. Character models have been mentioned, but these do not react in real time to a user. For that, we need a multi-agent system.

Do colloquial.

↪ characters as intelligent Agents?

### 1.2.2 Intelligent Agents as Characters

Story worlds are usually populated with characters. Interactive story worlds such as games contain characters with very basic scripted behaviour. At even the highest level of game character simulation, AI techniques are usually used to govern basic behaviour such as movements and actions. Governing character behaviour to fit within the context of a narrative is a more challenging problem. This section examines different approaches to tackling this challenge.

### Planner-based Systems

The most prevalent approach to the generation and management of plot in interactive narrative is to use planners. With a planner, an author sets the goals for the story (certain situations that they would like to see happen), and the planner tells the character agents what to do to make sure these "story goals" are achieved. When a player that is interacting with the system takes an action that compromises the story goals, the planner must re-plan to make sure the goals can still be achieved, by restricting the actions of the player or intervening in some other

Refs ?

manner.

(Young, 1999) argues that planners are a good method for regulating plot, later creating the *Mimesis* architecture for integrating a planner with character agents in an interactive game environment, Young et al. (2004). Young describes how narrative systems must re-plan when a player makes narrative-breaking actions, by either restructuring the narrative mid-story (*accommodating* the action) or preventing the action from executing (*intervening* on the action).

Given its influence over subsequent approaches to interactive narrative generation, it is worth looking at the *Mimesis* architecture more closely. It is designed to integrate into the *Unreal Tournament* game engine, and has five components: the *mimesis controller*, the *story planner*, the *discourse planner*, the *execution manager* and the *MWorld*.

Figure 1-6 shows how these components work together to form the *Mimesis* architecture. Once an author has created the pre-defined libraries of actions needed by the story planner, the following steps are taken to determine the course of the story:

1. All the components connect to the *Mimesis Controller* (MC) via socket connection. It then acts as a message router.
2. The game initiates a plan request containing the state of the game world, a list of possible actions, and the goals for the plan.
3. The *story planner* responds with a *story world plan*, a data structure that describes the actions (selected from the list) that must occur over time in order to meet the plan request's goals.
4. Once the story world plan is created, it is sent to the *discourse planner*, along with a list of actions that can occur in the game engine (such as camera movements, voice-overs and background music). The discourse planner then creates a sequence of these actions that best fit the story world plan.
5. The discourse planner then sends the narrative plan to the *execution manager*, which builds a directed acyclic graph (DAG), where the nodes are the actions within the plan and the edges are the temporal constraints between the orderings of the actions. The execution manager removes nodes from the DAG in order, sends the node's actions to the game engine, and updates the graph.
6. The *MWorld* is essentially the environment in which the story occurs, consisting of the game engine, coordination code, and class definitions for actions, and discourse planners. It is the *MWorld* that receives actions from the execution manager to be executed, and executes these actions in the game engine.

*Mimesis* uses DPOCL (Decomposed Partial-Order Causal-Link Planner) Young et al. (1994) plans for its story planning. DPOCL plans are composed of *steps* (the plan's actions), *ordering constraints*, *decomposition links* describing the hierarchical structure of a plan, and *causal links* between pairs of steps. It uses *refinement search* Kambhampati et al. (1995) as its plan reasoning process, searching through the space of possible plans represented as a directed graph, with each node in the graph being a plan or partial plan. *Mimesis* specifies the initial planning problem for DPOCL using the current and goal states of the story.

## Drama manager-based Systems

Carnegie Mellon University's OZ project Mateas (1999) uses a *drama manager* to structure its narrative, which observes the actions occurring in the storyworld and "directs" its characters to conform to shape the story.

Mateas and Stern's *Faade* has players interact with the characters of the story through natural language. In this game, the player attends the party of a young couple (Grace and Trip) celebrating their wedding anniversary. As the course of events unfold however, the player learns that all is not as happy as it seems.

The player interacts with the characters by typing in natural language sentences, to which Grace and Trip respond. Though the characters are implemented through agents, the story is controlled using a drama manager. In all, their system consists of using NLP, a novel character authoring language and a novel drama manager to create an interactive narrative.

Several custom-designed languages were used to create the game, including a language called 'A Behaviour Language' (ABL) for the agents and a special language for the sequencing of the beats. ABL represents situations as character goals, maintaining a tree of all the active goals and behaviours that are happening at any time.

In *Faade*, the smallest unit of narrative action is called a *story beat*, taken from McKee's book on authorial style for screenwriters (McKee, 1997). The simulation constantly monitors what the user is doing and how it may lead from the current story beat to another. Story beats have preconditions and effects on the state of the narrative, so it is the drama manager's job to work out when it makes sense to initiate a certain beat.

'Beats' have a very fine granularity, with 200 or so updating every minute of the simulation. They consist of a set of ABL behaviours, which advance the narrative yet still allow interaction to change to other beats. Only one beat can be active at a time.

A beat can have 5 types of goal:

1. transition-in: characters express their intentions
2. body: a dramatic question/situation is posed to the player
3. local/global mix-in: react to the player before end of the beat
4. wait-with-timeout: wait for the player's reaction
5. transition-out: final reaction to the player's action in the beat

A beat goal is a series of steps for an agent to perform, which can be:

- staging (where to walk to, face)
- dialogue to speak
- where and how to gaze
- arm gestures to perform
- facial expression to perform
- head and face gestures to perform

- small arm and head emphasis motions triggered by dialogue (head nods, hand flourishes)

As an example, there is a behaviour called "Fix\_Drinks", which specifies a sequence of agent behaviours where the characters Grace and Trip have an argument while Trip asks the player what they would like to drink. If the player decides not to go along with the beat (in

parenthetical beat

is this a quote.  
Single quotes might be preferable.

Worth a screen shot?

NLP?  
is undefined.

is the order significant?

embedded underlined

insert a gap. Doesn't need to be needed under #5

increase gap.

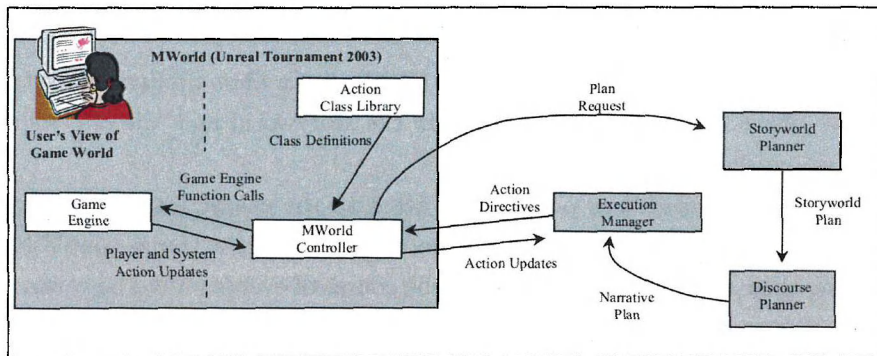


Figure 1-6: The Mimesis architecture, from Young et al. (2004)

A key feature of Mimesis is its handling of user actions which potentially interfere with the story plan, making its goals unachievable. Its two strategies for mediating in such situations are *intervention* and *accommodation*. In the intervention strategy, Mimesis simply prevents the user's action from having any effect in the game world. With accommodation, Mimesis replans the story events, restructuring the plan so that the interfering actions are taken into account and worked around.

Cavazza et al's *I-Storytelling* system Cavazza et al. (2002) implements Young's architecture with ideas from Barthes' narrative units, using characters with Hierarchical Task Networks (HTNs) to generate its stories. Each character has a main task, which is divided into subtasks to create a task hierarchy, with each task node having pre- and postconditions. The story emerges from the outcomes of each character's plans, and the narrative structure as a whole is not planned.

(Riedl et al., 2003) describes further development of Young's architecture, allowing the story author to create plans for the overall narrative in addition to its characters, in order to have more control over narrative structure. Rather than using a narrative model, this version of Mimesis models the player to track their expected level of suspense while interacting with the story. Like the *I-storytelling* system, its plans are hierarchical, using the Longbow hierarchical partial-order causal link planning system Young et al. (1994).

A disadvantage of these planner-based systems is that they require the story author to think in a planner-oriented manner. They must consider the goals of both the story and the character, plans to achieve these goals, and re-planning when goals are not met or when situations change. This is a drastic change from the usual story writing methods of authors, where the focus is on structure, plot, themes and characters. Though graphical user interfaces such as Mimesis' Bowman system Thomas and Young (2006) could be used to assist planner-driven story authoring, a complete shift in creative workflow is still required, making them inaccessible to non-technical writers.

repetition

insert cite after Riedl Cavazza

\*ref

"characters with HTNs?"  
meaning is unclear.  
"characters with behavior described by HTNs" possibly

too obvious

Textual cite

parenthetical cite

parenthetical cite

plan-driven?  
this approach?



this case, by not choosing a drink}; then the beat will be aborted and replaced with another.

Façade has become popular as a game outside of academia, with playthroughs of the game reaching millions of views on Youtube. This shows the promise of interactive narrative as being a unique and engaging new form of entertainment. Unfortunately, no other implementation of interactive narrative seems to have captured the public imagination since the release of Façade.

Façade's popularity seems to reinforce Crawford's assertion (section ??) that interactive narratives must be social in nature. The gameplay comes entirely from the conversations and interactions between Grace, Trip and the player. Much of the excitement comes from the social consequences of certain conversation paths or actions. By modelling characters as agents, Mateas et al have created a truly interactive experience. However, by also using a drama manager to manage the agents, they have used these agents to tell a story.

From the analysis above, we conclude that using social institutions to govern the actions of character agents allows for more flexibility in the agents' actions. Also, by specifying these institutions as story tropes, we overcome the authorial difficulty of planner-based systems. Script writers do not need to be familiar with planners or hierarchical task networks in order to create interactive fiction with our system.

How might these agents be made more convincing? Outside of writing rules for their behaviour consisting of character goals and beliefs, how might an author create truly unique and idiosyncratic characters? To address the question, I next examine different types of emotional models in psychology, and how each might be used to model characters as agents.

### Social Norms

Versu Evans and Short (2014) is an interactive drama system that uses a multi agent system as characters. The characters' actions are coordinated with *social practices*, which describe types of social situations and is described by the authors as a successor to the Schankian script. These social practices are implemented as reactive joint plans, which agents can choose ~~whether~~ to participate in or not. Rather than directly telling the agents what to do, these social practices merely *suggest* courses of action, leaving each agent to decide for itself what to do based on its individual goals.

The authors decide against using a drama manager to control the agents' actions because they want to take the *strong autonomy* approach to agent governance. This means that they prefer to give each agent some degree of autonomy by allowing it to make the final decision on which course of action to take, rather than blindly following a drama manager. Suggesting actions with social norms achieves this goal. Rather than describing typical story events in terms of social norms, however, in Versu the social norms *are* the story. The gameplay revolves around the avoidance (or purposeful subversion of) awkward social situations.

Each character has a role, which is governed by a social practice. For example, a *greeting* practice involves characters with the *greeter* and recipient roles. The greeting practice would tell the greeter in which manner they are to greet the recipient, and the recipient how to respond. It is noteworthy that these actions are merely suggested, and not enforced.

Tropes  
undefined  
as yet.

I think  
this should  
be moved  
out of the  
lit review.

redundant

not sure the  
logic holds.  
This is your  
hypothesis.

This could do with a diagram

It manages the frame-problem by organising related fluents in a tree structure. A single update of the right branch can change a set of related facts en masse

Exclusion logic Evans (2010) is used to describe the social practices of the system. Exclusion logic allows the description of tree structures and includes an exclusion operator. For example, a description of a character called "Brown" is shown in listing (??). It describes the building up of character attributes as a tree structure.

Exclusion logic aims to address the frame problem. The frame problem is the uncertainty around whether predicates that change over time (fluents) change other predicate values. It aims to address this through use of an exclusion operator ("!"). Listing (??) shows an example of the exclusion operator in use. The example specifies that an agent can have only one gender. The Praxis language implementation of the exclusion logic has a type checker which ensures that no character can have multiple genders.

Exclusion logic's exclusion operator allows an author to express the fact that a variable can only have one value. For example, if 'the "Brown" character changes location from the dining room to the kitchen, *brown.in!dining\_room* is terminated when *brown.in!kitchen* holds.

Versu takes the *constitutive* view of social practice, as opposed to the *regulative* view. This means that rather than restricting an agent's possible actions based on its permissions and obligations, they participate in a certain social practice by taking an action. Their actions are only restricted by what is possible in the story world, and what the agent desires to do. This way, agents can choose whether or not to take part in certain social interactions.

Many of the components we aim to have in our story telling system appear in Versu: the use of social norms to gently encourage story-conforming behaviour rather than demanding it, and the use of formal logic to determine which behaviours are possible. However in Versu the social norms *are* the story, rather than describing the story components that invisibly govern the behaviour of characters. In order for this kind of governance to occur, an institution-based solution is preferable, based on events, agent actions and standard deontic logic. Because character actions are constrained by the structure of a story, a *regulative* view of social practice is more suited to the expression of story components as social norms.

Many Much of the advantages of using exclusion logic can be gained by using an institution-based approach. Non-inertial fluents can be used to ensure that variables can only ever have one value. Standard deontic logic is enough to provide the rest of what is needed.

### 1.2.3 Modelling Narrative with Logic

Although most recent research focuses on the use of planners to manage the drama in a story, there is also much interesting work which makes use of formal logic to model narrative. Though often used for the generation of linear story text, it is increasingly being applied to non-linear narratives as well. Logic-based approaches are generally based on either temporal logic variants or some kind of linear logic.

Ceptre (Martens, 2015) is a language for modelling generative interactive narratives using *linear logic*, a formal logic designed to describe resource usage.

A Ceptre story begins with an initial state  $\Delta_0$ . Each state iteration  $\Delta_{i+1}$  is examined repeatedly, and a subset  $S$  of it is updated with rule  $r$ . The next state,  $\Delta_{i+1}$ , has the subset  $S$  replaced with  $S'$ , the new subset with the consequences of the applied rule  $r$ .

Code snippets?

Can you add more fluents that would be in the same frame. If you add any related fact Ref !! (Kant) cy. He can be in the living room watching TV. This gets deleted with the move.

\* Ref !!!

'Delta' or  $\Delta$

The rules are specified using the combination of logical statements with two operators: \* (tensor) and -o (loli). The tensor operator is used to concatenate statements, while the loli operator expresses state transitions in the form  $S -o S'$ . The rules use *replacement semantics*, which means that everything from state  $S$  will disappear unless stated to be in state  $S'$ . A \$ operator is used to mark facts in  $S$  that the author wishes to remain in  $S$  without explicitly stating so.

Listing ?? shows an example from Martens (2015) that describes a “murder” rule and its consequences.

In this case, four instances of the “anger” predicate with the same arguments has a significant meaning: a character’s emotion is treated as a resource. The fact that *anger CC'* appears four times means that a character is *four times* as angry at another character. Depending on how many times the “anger” statements appear in the new state, this anger can rise or fall at the next step in the story. In this case, the emotion is not only treated as a pre-condition, but also as a resource. These resources can also be specified through the addition of a number to the name of the resource.

Ceptre introduces a *stages* feature that allow authors to structure a program through the use of independent components. A stage is a unit of computation that runs to quiescence, meaning that it terminates ~~once~~ <sup>when</sup> no more rules are able to fire. At this point, another stage may commence.

The central motivation behind Ceptre’s design is its ability to use “proofs as traces”, or *computation as proof search* Hodas and Miller (1991). Ceptre uses a *sequent calculus*, where a sequent  $\Delta \vdash A$ ,  $\Delta$  is a state, and  $A$  is a goal formula. If a complete proof tree can be formed with that sequent as its root, then the sequent can be said to be *derivable*. Thus Ceptre takes a sequent as input and creates a proof as output, declaring failure if a proof cannot be created. Ceptre looks at the left side of the sequent, using *forward chaining* to choose which rules to try in order to reach the goal formula.

A key feature of Ceptre is its representation of resource management in stories. Rules are able to either produce or consume resources. This has interesting implications for the representation of causal structure in linear logic. For example, if two rule applications consume different sets of resources from the same state, they are occurring concurrently and independently. However, if one rule produces resources that are consumed by another rule, then these rules have a causal, dependent relationship.

This modelling of gameplay as proof search is similar to the technique we use in section ??, where we use Interval Temporal Logic and Kripke structures to theorem-prove certain narrative states. The difference is that the system we describe is more concerned with representing different temporal relations, whereas Ceptre’s focus is on resource management within a game.

example needed.

possibly  
outside  
Lit.  
Review.

Smarter easier  
to manage if  
you minimize  
formal references.

#### 1.2.4 Character Modelling

##### Characters with Emotional Models

##### Emotional models

Usually it would seem odd to want to model emotion as part of a computational process. Emotion is such a seemingly irrational set of behaviours that they are easy to dismiss as 'human imperfections'. However, as Gratch and Marsella (2004) observe, emotions may have a useful role to play in communication, so long as they are displayed at appropriate times.

For example, anger prepares the human body to fight by increasing the manufacture of adrenaline. Fear similarly triggers the 'fight or flight' response, alerting the senses for danger and preparing the body to react.

In order to model human emotions using agents, we must first find a suitable psychological model to use. Marsella et al describe three main types of emotional model:

1. **Discrete** emotional models, which claim that humans have a set of innate, pre-defined emotional states which people may enter and leave.
2. **Dimensional** models of emotion, describing the spectrum of emotions as being points somewhere in continuous space. Implementations typically use two or three dimensions for simplicity.
3. **Appraisal** theories of emotion take an agent's mental processes into account. Their emotional state is derived from whether or not their goals have been achieved, and what effects current events are having on their circumstances, for example.

##### 'Basic' emotions

Ekman first made a case for discrete, biologically-determined emotions, based on evidence from research into facial expressions (Ekman, 1992). He describes emotions as being *basic*, in two senses of the word: *i.* that there are a number of distinct emotions, each with its own different characteristics, and *ii.* that these emotions were developed through evolution for specific functions.

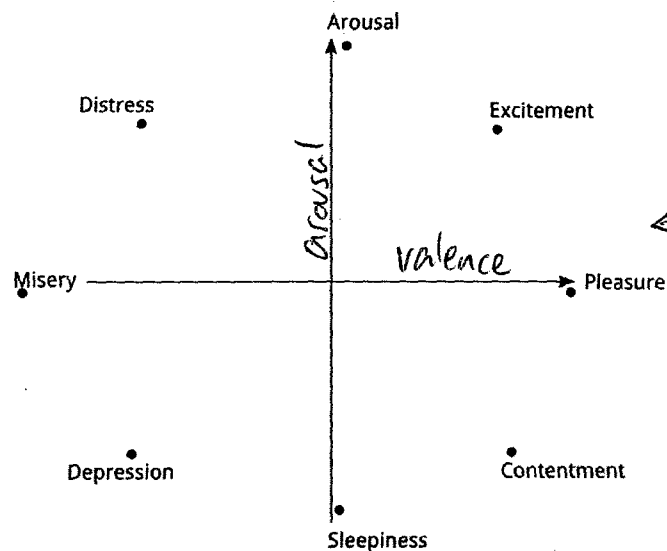
Ekman argues that these evolved emotions share nine characteristics:

1. Distinctive universal signals
2. Presence in other primates
3. Distinctive physiology
4. Distinctive universals in antecedent events
5. Coherence among emotional response
6. Quick onset
7. Brief duration
8. Automatic appraisal
9. Unbidden occurrence

These characteristics are shared by all of the 'basic' emotions as observed in humans and primates.

Don't emotions also  
help to create more  
believable agents?

production



← Axes aren't labeled.

created

Figure 1-7: Russell's circumplex model of emotion

Discrete models of emotion suggest that there is a neural basis for emotion. For example, Armony et al describe how the amygdala in the brain is responsible for conditioned fear responses, and create a neural network to model it (Armony et al., 1997).

Using a discrete model of emotion for agent-based characters would be relatively simple. Each basic emotion could have its own distinct set of behaviours as postconditions, and triggering circumstances as preconditions.

is that really a postcondition? or then as a 'consequence'

However, a more fluid approach could be useful when modelling emotions with agents. It would be impossible to say that an agent is *angry and approaching furious* using a discrete theory of emotion. Nuanced levels of emotion and even combinations of several emotions add an extra level of texture to a character. Dimensional and appraisal theories of emotion address this challenge.

### Russell's circumplex model of emotion

Russell's circumplex model of emotion is a well-known dimensional model (Russell, 1980). In this case, the dimensional variables are *valence* (how agreeable or otherwise a situation is to an agent) and *arousal* (how excited an agent is).

Russell's original paper proposes a model similar to that shown in figure 1-7, where the  $x$  axis is a person's valence level and the  $y$  axis is their arousal level. He argues that the full range of human emotions lie as points along these axes. Eight such examples are shown in fig. 1-7.

This model is very easy to adapt to human-like agents. Ahn et al. (2012) adapt this model by adding a third dimension, dominance, to create conversational agents in a 3D environment. This 'dominance' dimension was first proposed in Mehrabian and Russell's original work (Mehrabian and Russell, 1974), but later removed due to being perceived as the consequences of the *effects* of emotion (Russell, 1980), rather than being a component of emotion

itself. Like Ahn et al, I found it useful to add the dominance-submission dimension, and so left it in my emotional model. This is the approach I take in creating my Punch and Judy simulation, and so it is described in more detail in section ??.

lit review?

### Appraisal theory

Appraisal theories of emotion lend well to simulation with agents, due to their taking a person's beliefs, desires and intentions into account with respect to external events. Emotions arise when an event occurs and a person internally *appraises* its consequences with respect to their beliefs, desires and intentions. This fits well with the popular BDI architecture for intelligent agents.

Different methods of appraisal may be used in order to produce emotions. Gratch and Marsella use decision theoretic plans (Gratch and Marsella, 2004), but other approaches could include reactive plans, Markov-decision processes, or detailed cognitive models.

Though the Punch and Judy simulation described in section ?? uses a dimensional model of emotion, an appraisal-based model would be worth investigating due to its tight coupling with belief desire intention psychological models used in agents. I describe my intention to explore this area further in section ??.

use textual cited

lit review?

fun intended?

### 1.2.5 Discussion

Through this literature review, we have clearly identified these main issues in need of attention:

#### Story authors do not want to think in terms of goals

The current dominant paradigm in interactive narrative creation is to use planners to "plan" a narrative, and re-plan when a user interacts in an unexpected manner. This does not align well with non-programmer story authors, who are likely to be unfamiliar with planning systems. In order to create these stories with planners, they would have to think of the story in terms of story and character goals.

#### Most narrative systems use outdated, inflexible story models

There is an over-reliance on narrative formalisms such as Propp (Propp, 1968) even in recent narrative generation systems. Neither narrative nor AI research have produced a formalism for narrative components that has endured. This is likely because Propp's formalism is "good enough", and has worked for most researchers that have used it. There is also likely to be a network effect, where Propp has become the formalism that most researchers have heard of, and therefore the one that they end up using.

Though there have been attempts to create more expressive formalisms as described in section ??, none have been expressive enough to overcome Propp's "good enough" properties. In order for a story model to significantly improve upon Propp, it should add features that it and other existing formalisms lack. We identify these features as:

- A means of *abstraction*



- Conceptually *simple* enough for non-programmers to grasp
- A library of re-usable *examples* for authors and researchers

**A Means of Abstraction** Current narrative formalisms lack a means of *abstraction*. Propp, for example, describes events in stories that all occur at the same level of abstraction. This means that one is limited to describing events that occur one after another, or in parallel, but not events that contain sub-events.

For example, suppose we define a “Quest” component of a story, which describes a sequence of events that occur (such as the hero leaving home and then defeating a monster). It is easy to think of other story components that could contain this “Quest” component, such as “Hero’s Journey”, “Rescue the World” or “Rite of Passage” story components. These components could themselves be used as part of other components. This gives us a means of creating abstractable, reusable story components. Of all the story models described in section ?? none have any means of abstraction such as this.

The example we just described already hints at the use of story tropes that contain other tropes. This is the means of abstraction and re-use that we use in our system, described in section ??

*Still undefined.*

**Conceptually Simple** Story authoring tools are user interfaces which are used to write fiction. In the computing world, user interfaces are often simplified through the use of an *analogy*. For example, a *Desktop Environment* is a graphical user interface analogy created by Xerox in the 1970s. Where computer users would previously have had to learn and master a complicated command-line interface, the desktop environment metaphor presented an interface that resembled something that users were already familiar with: the top of a desk, with icons representing files spread out over a “desktop” surface.

When creating a story model, it is easy to fall into the trap of creating something complicated but arbitrary. As with Vonnegut’s “Shapes of Stories” metaphor (described in section ??), sometimes the simplest explanations are the easiest to grasp and most enduring. We believe that our “trope” analogy for modelling stories (section ??) is an elegant, expressive and accessible way of describing parts of stories using a formalism.

Most story authors are already familiar with the concept of tropes. When questioned about their familiarity with the idea, all the participants at an interactive fiction meetup responded that they were familiar with the “trope” concept (sample size 19, see section ??). This means that tropes are a suitable analogy for the creation of a new narrative formalism.

**A Library of Re-usable Examples** Part of the problem of existing formalisms is that story authors that use them need to write their own story components based on a formalism’s description. Though these formalisms are usually described in papers with one or two examples, any story author would have to create their own formal descriptions, even for commonly-occurring story components such as quests or *The Hero’s Journey*. What is needed is a library of pre-existing formal descriptions of story components that authors can easily copy and paste into their stories.

As the concept of *tropes* is one that is already well-known to story authors and consumers of media, it is easy to find many examples of them in reviews of books, films and computer games. In fact, there is a whole website called “TV Tropes”, which is dedicated entirely to the description of tropes that recur throughout different types of media. This website takes the form of a wiki, with a very active community who contribute trope descriptions along with lists of the media they appear in.

For example, a trope called “Karma Houdini” has the following description:

“ The character has done a number of things that deserve a karmic comeuppance, most importantly things that caused harm to the innocent. But when the time comes for the hammer to fall, that’s not what happens. At least, not on him. He doesn’t get what he deserves. Instead, he gets away scot-free. And he might even have reversed the Humiliation Conga that was being planned for him.

This is it. This is all there is to the story. The show is over. The book is finished. The author isn’t going to write any more. The Word of God has been spoken. The guy has become a Karma Houdini. ”)

The site lists many examples from film and literature, including:

- **Treasure Island:** Long John Silver escaped scot free with a chest of treasure, and was never caught. Not bad, for a month’s murder and betrayal.
- **The Talented Mr Ripley:** Villain Protagonist Tom Ripley killed some people to assume a new identity and enrich himself thoroughly. In the sequels, he killed to protect his new life, and sometimes as favors for others. He never faced justice.

The *TV Tropes* website is a Wikipedia of sorts for tropes. Since there are already so many examples listed on its website, it reduces the work needed to create a library of reusable formal descriptions of tropes. We have created such a library, which is described in section 2.

These three omissions (means of abstraction, conceptual simplicity and a re-usable library of components) from existing narrative systems are addressed by our trope-based approach to interactive narrative generation, which is described in further sections of this thesis. Section 3 describes the trope concept further, with examples. Section 4 explains how these tropes are used to govern agents in a multi-agent system. We describe TropICAL, our domain-specific language for tropes, in section 5. Section 6 describes our creation of a library of tropes and section 7 describe the application of our *tropes* concept to legal policies. The thesis concludes with an evaluation in section 8 and conclusions in section 9.