Never-ending Storytelling with Discrete-Time Markov Processes

Yutu Liu, Ergun Akleman and Jianer Chen
Texas A&M University

Abstract

This paper presents a theoretical framework to create never-ending stories. Our theoretical framework is developed using discrete-time stochastic processes with a Markov property. To demonstrate the effectiveness of our theoretical framework, we have developed a visual storytelling system that helps users to interactively create narratives. This preliminary visual storytelling system can help users to create never-ending narrative segments in cartoon form.

1 Introduction

Visual storytelling is one of the areas in visual aesthetic that is not well understood. Visual storytelling still requires a lot creativity and hard work. The major portion of any movie production process is spent for the development of stories and storyboards. To improve the visual story development process there is a need for new theoretical approaches as well as new tools and techniques. It is, therefore, there is a recent interest in interactive storytelling [12]. Mateas and Stern developed one of the first interactive storytelling software called Facade [11]. Iurgel et al. developed IRIS, another interactive storytelling system, which is based on a hidden Markov Model [8]. There also exists commercial web-based interactive storytelling software such as Xtranormal [17].

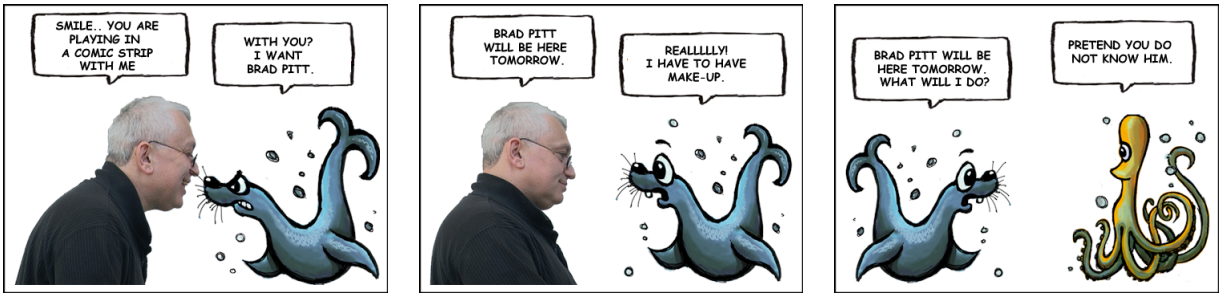

Figure 1: An example of a short narrative segment created with our approach by using our system.

This paper introduces a theoretical framework that can be helpful for the development of such tools and techniques. Based on this theoretical framework, we have identified that discrete-time stochastic process with a Markov property can be useful for the development of tools to create never-ending stories. We call these stories never-ending because they never really finish or complete. They can continue indefinitely. Never-ending stories are most common story types that are used in soap operas and television sitcoms. Even news and social relationships can be considered never-ending stories. Using Markov processes, we have developed a proof-of-concept system to create never-ending visual stories. This preliminary system shows the potential of our approach for story development by helping users to create unusual and never-ending story segments. Fig 1 shows a semi-automatically created story segment using our system.

2 Theoretical Framework for Narration Modeling

Narratological analysis was started in the 1920s by Vladimir Propp [15], who developed a grammar covering a restricted corpus of Russian folktales. Propp's analysis was used in some early story-telling programs in

Artificial Intelligence such as ([13] with limited results. Propp's theory was substantially refined in the 1960s by Barthes, Greimas and Bremond [1, 7, 2]), when a distinct discipline called "narratology" emerged.

Narrative theoreticians agree that there are at least two levels in any narration: Some events happen and these events are related in a certain way. Although there exist various terminologies used by different researchers [9], these two levels of a text can be identified by two questions:(1) What is told and (2) How is it told? In the most widely used structuralist terminology, the answer to the "what" question is called a *story* and the answer to the "how" question is called a *discourse* [3].

For the purpose of this paper, we use an earlier and narrower approach that is first introduced by Aristotle and refined by the novelist and critic E.M. Forster in his *"Aspects of the Novel"* [6]. Aristotle and Forster categorize a narrative in two levels as (1) Story: A chronological sequence of events (at least one event) (2) Plot: A causal and logical structure which connects events (at least two events).

Forster's well known example for story is *"King died and then Queen died"*. Since this is a sequence of events, it is not a plot. On the other hand, *"King died and then Queen died of grief"* is a plot since adding *"of grief"* creates a causal relationship between the two events. Forster's categorization is helpful for the development of theoretical framework. It is specific enough to allow just one event as a story and it is general enough to allow indefinite number of events as a plot without any specific structure.

The major problem with Forster's categorization is that it does not provide a definition for an event. For the development of a theoretical framework, there is a need for a definition of an event. We define an event as a causal and logical structure that connects statements. In other words, a transitional cause that describes a change from one state to another is an event. For instance, *King is alive* and *King is dead* are statements and *alive* and *dead* are states. The two different events *"King died* and *"King is killed* can be written as a transition from *alive* state to *dead* state by changing the cause as the following.

$$\textit{King died} \quad \equiv \quad \textit{King is alive} - \textbf{died} \rightarrow \textit{King is dead}$$
$$\textit{King is killed} \quad \equiv \quad \textit{King is alive} - \textbf{killed} \rightarrow \textit{King is dead}$$

Based on this discussion, we provide the following definitions by extending Forsters categorization.

• **Universe:** The universe consists of a finite *character space* $\mathscr{C}$, a finite collection $\mathscr{S}$ of finite *state spaces*, and a finite *expression space* $\mathscr{X}$.

• **Character space:** The character space consists of a finite number of *characters* $C_0, C_1, \ldots, C_n, \ldots, C_{N-2}, C_{N-1}$. Each character can be a person, an animal, or an object. Each character is associated with a collection of state spaces and with an expression space.

• **Statement:** Each *statement* is made of a subset $\mathscr{C}'$ of characters in $\mathscr{C}$, where each character C in $\mathscr{C}'$ is described by a tuple $(C, S_1, \ldots, S_r, X)$, where each S_i is an element from one of the state spaces associated with the character C, and X is an element from the expression space associated with the character C.

• **Event:** An *event* is a causal and logical structure that connects two statements.

• **Story:** A *story* is a chronological sequence of events. There should be at least one event.

• **Plot:** A *plot* is a causal and logical structure that connects events. There should be at least two events.

Note that we treat everything existing in the universe as a character. Some characters contribute to the story while others do not but act like props or background elements. The decision of who and what can be a contributing character depends only on the user. Therefore, in our framework a character can be an animal, an object or even a background element. If the user chooses so, an object or a background element can contribute to the story. For instance, if there are two statements that follow each other as "House is yellow" and "House is black", then these two statements can be causally connected as "Yellow house is painted to black" or "Yellow house is burned and becomes black". Then, the result is an event and defines a story. A a result, the house becomes a contributing character to the story. If no explanation is given by the user, the house is still a character but does not contribute to the story.

Another important property is that each character can only be associated with a restricted collection of state spaces. The reason behind this is that each character is fundamentally different. For instance, a house can have different colors as states. On the other hand, a person can have emotions as states. However, the

states associated with characters do not always have to be logically consistent. For instance, a user must still be able to choose state spaces that are usually used for people as state spaces for an object. Such usage allows to create metaphorical statements that can spark people's imagination. For instance, sky can be angry, house can be dead; nose can be tired.

The separation of statements from events, story and plot allows us to develop simple mathematical models to create statements automatically. The model needs to provide new states for each character from all the previous and current states of all characters in the universe. It is possible to use a wide variety of mathematical approaches to develop such models. The only consideration is that the model must provide the user with friendly control over the personalities of characters, and results should be corresponding to our actual experience.

To create statements automatically, we have considered several mathematical approaches such as genetic algorithms, expert systems, plot-based methods, directed graphs and trees. All these approaches have potential to provide different ways to stories. However, we have identified that Markov processes are the most suitable mathematical approach for never-ending storytelling. In the next section, we provide a set of particular expectations to use Markov processes for storytelling.

3 Discrete-Time Stochastic Process with a Markov Property

A collection of states associated with a character is used to represent the states of the character in a specific aspect. For instance, the following are five common aspects for which a character may have states, each of them makes a state space in our construction:
• *emotional states* such as anger or happiness,
• *physical states* such as walking, sitting or running,
• *internal physical states* such as sleepiness or tiredness;
• *states for drawing balloons* i.e. talking and thinking; and
• *existentialist states* such as live, dead or ghost.

Note that states in these aspects may exist relatively independently, and that a character can have a state in each of the above aspects at the same time. For instance, a "living" person can be "angry" while "walking" or a zombie i.e. a "dead" person can be "angry" while "walking".

As we defined early, each *statement* consists of a subset $\mathscr{C}'$ of characters in $\mathscr{C}$, where each character C is described by a tuple $(C, S_1, \ldots, S_r, X)$, where each S_i describes the current state of the character in a particular aspect, and X is the current expression state of the character C. Each *event* gives a transition from one statement to another statement.

One of the most important problems in our development is how the events are made, i.e., how the statements transit from one to another. Different rule sets for events can lead to different mathematical approaches. We have identified that we expect the following rules from our never-ending storytelling processes.
• **unpredictability.** The story should not be predictable. In particular, the same statement should not always result in the same following statement;
• **logical continuity.** Many states of a character should last for a while. For example, a person who is currently happy is probably still happy in the the next moment. On the other hand, certain other states do not last very long. For example, surprise should not last long;
• **personality consistence.** Each character should have its own personality, which should have a significant impact on how the character behaves. For example, a happy person is probably happy most of the time, and a sad person may rarely be happy;
• **environment impact.** The states of a character should also be dependent of other characters' states. For example, a person who is usually happy may become scared if he is facing a rude and angry person. Note that the environment impact on a character also depends on the personality of the character: a very social

person may be influenced easily by the environment, while a less social person may be less easily changed by the environment.

For this particular rule set, the Markov chain model provides a very useful tool. To illustrate how the Markov chain model can be used, we will first assume that each character is associated with only one state space and an expression space. We will explain how this simple model is extended to a collection of more than one state space.

Let C be a character, and let $\mathscr{S} = (S_1, \ldots, S_n)$ be the state space associated with C. A Markov chain representing the state transition distribution for the character C can be depicted as a directed graph G_C in which each edge is labelled by a real number p, $0 \leq p \leq 1$. Each vertex in G_C corresponds to a state in $\mathscr{S}$, and a directed edge (S_i, S_j) labeled by $p_{i,j}$ gives the probability by which the state S_i transits to the state S_j. Note that the sum $\sum_{j=1}^{n} p_{i,j}$ should be equal to 1. Alternatively, the Markov chain can be represented by a matrix P_C, called the *transition matrix*, in which each row adds up to 1.

To illustrate our idea more clearly, we assume that the state space $\mathscr{S}$ consists of four states: happiness, sadness, anger, and fear. The following gives the transition matrices for two characters:

C_0	happiness	sadness	anger	fear
happiness	0.75	0.05	0.1	0.1
sadness	0.1	0.4	0.3	0.2
anger	0.05	0.4	0.4	0.15
fear	0.1	0.35	0.1	0.45

C_1	happiness	sadness	anger	fear
happiness	0.5	0.2	0.2	0.1
sadness	0.05	0.6	0.15	0.2
anger	0	0.3	0.6	0.1
fear	0	0.6	0.1	0.3

Note that the transition matrix is defined based on the personality of the corresponding character (thus, it can also be called the *personality matrix* of the character). In general, a character may have a preferred state. For instance, a happy person tends to be happy. Thus, from a state, the happy person has a higher probability to be happy in the next statement. The personality difference between two characters can be seen by comparing the personality matrices of the two characters. For example, by comparing the above two personality matrices, we can observe that the first character will more likely be in happy state than the second character, therefore the first character seem happier than the second character.

Since elements of each row in the transition matrix P_C must add up to 1, when users define these matrices they may not be able to make each row add up to 1. This problem can easily be solved by letting users enter any number by assuming that the higher the number, the higher the transition probability. Then, the matrix can be constructed from user entries by normalizing each row. However, we point out that the personality matrix of a character does not have to be made by the users. It is also possible to provide a number of prescribed matrices. There can be other strategies. For instance, the users can also be given a list of queries on each aspect of personality of the corresponding character (such as "is the person happy?", and "is the person easily angry?"). The user can simply give a score in the range of 1 to 10 for the aspect. Then the system can make the personality matrix for the character based on certain fixed rules. There are two extreme cases: (1) the user is not involved at all in the definition of the personality of the character, and the user assigns an existing personality matrix for the character; or (2) the user can provide all details for the personality matrix and give the character a special (perhaps unusual) personality (such as "happy and easily angry person", or "sad and easily angry person").

The most important property of transition matrices based on Markov chains in this paper is that the long term behavior of a character is predictable. For a character C with a transition matrix P_C, we have $\lim_{m\to\infty} \inf P_C^m \longrightarrow \Pi_C I$ where Π_C is the left eigenvector of the transition matrix P_C for the eigenvector 1 and it is called the *stationary distribution*. In other words, each element of Π_C tells us what fraction of time the character C stays in a certain state. In other words, we can design characters who can be mostly happy, mostly sad or mostly angry. Although the characters' next state will be random, their behavior will be predictable in long term. More importantly, the users will be able to define the long term behavior by

creating the matrix.

It is also important to note that some transition probabilities in the personality matrix cannot be freely chosen. For instance, unlike some exceptional cases such as the movie Illusionist (http://www.theillusionist.com/), a dead person does not become alive again. So, the transition probability from dead to live must be 0 and the transition probability from dead to dead must be 1. Surprise never lasts for a long time, so transition probability from surprise to surprise should be small.

Time-homogeneity is not always a desired feature in narration. Certain plots such as transformation or maturation [16] require that the character's behavior change with time. For such cases, it is possible to interpolate two matrices in time. In other words, a character who is usually sad can be transformed to a character who is usually happy by interpolating its transition matrices over time.

3.1 Feedback Mechanism and Sociability

The personality matrices are useful for development of individual characters who are not affected from other characters. However, the characters do not exist in vacuum. Their behavior must change based on the states of other characters. This is mainly because the narrations are representations of interactions among humans who are really social creatures. For instance, a person who is usually happy may be sad if all the close friends are sad. Our model must incorporate these effects.

To incorporate these effects, we use a feedback from the environment. This feedback changes the personality matrix. In other words, if the environment is sad, the character becomes more likely to be sad.

Note that the current state of a character may have different impact on other characters. For example, a happy character may likely make other characters happy, but may (although unlikely) also make some characters sad. On the other hand, an angry character may make other characters scared but also may also make a reasonable group of other characters angry. For this reason, we introduced the concept of *impact matrix IM*, which defines the impact of the state of one character on another character. The following are two examples of impact matrices:

$C_0 \to C_1$	happiness	sadness	anger	fear
happiness	0.8	0.1	0.1	0.0
sadness	0.1	0.7	0.1	0.1
anger	0.0	0.1	0.4	0.5
fear	0.0	0.5	0.1	0.4

$C_1 \to C_0$	happiness	sadness	anger	fear
happiness	0.1	0.6	0.3	0.0
sadness	0.7	0.2	0.1	0.0
anger	0.1	0.1	0.1	0.7
fear	0.2	0.2	0.2	0.4

Again, each row of the impact matrix should add up to exactly 1. Therefore, for example, an angry person will not at all make other people happy, has a 10% probability to make other people sad, a 40% probability to make other people angry, and a 50% probability to make other people scared. Also note the differences in two impact matrices. The second impact matrix can be an example of two characters in competition. If one becomes happy, the other becomes sad. Of course, the impact is not commutative, i.e. impact of character C_0's impact to character C_1 may not be the same as the impact of C_1 to C_0. An example to this kind of impact relationship can be a successful character and jealous friend.

To implement how the state of a character C_0 is influenced by other characters, suppose that there are $n+1$ characters in the current statement: $\{C_0, C_1, \ldots, C_n\}$, we give the character C_0 a *sociability parameter*: s and a *closeness vector*: $V_0 = \{v_1, \ldots, v_n\}$, where v_i are the positive real numbers that add up to 1 and s is a positive real number smaller than or equal to 1. More specifically, suppose that in the current statement, the character C_i is in state S_i for all i, $1 \leq i \leq n$. Then the transition matrix of the character C_0 from the current statement to the next statement is given by $(1-s)P_0 + s(v_1M_1 + \cdots + v_nM_n)$ where P_0 is the personality matrix of the character C_0, and the matrix M_i for each i is a matrix in which every row is equal to the row in the impact matrix IM corresponding to the state S_i.

The sociability parameter and closeness vector of a character C_0 are closely related to the character C_0 and to the role of each involved character. A very social person has a relatively high value of s so his own personality P_0 has small impact to its next state. On the other hand, a less social person has a larger value s in its sociability vector so it is going to dominate its behavior. Moreover, the role of each of the other characters may also differ to reflect its importance to character C_0. A character who is dominating the statement (e.g., who is talking) can correspond to a larger v_i value in the closeness vector of the character C_0. On the other hand, an unimportant character (e.g., the background) may have very little to do to influence character C_0. Thus, the corresponding v_i value in the sociability vector of character C_0 should be small, or even can be 0.

Finally, we explain how the above system is extended when characters have a collection of more than one state space. To be specific, suppose that there are r state spaces $\mathscr{S}_1, \ldots, \mathscr{S}_r$ that are associated with a character C_0, such that the number of all possible combinations of the states is h. Then there are r transition matrices $P_C^1, \ldots, P_C^r$ associated with the character C_0, each is an $h \times r$ matrix. More specifically, the personality matrix P_C^i of C_0 describes the state transitions for the state space $\mathscr{S}_i$, in which each row corresponds to a combination $(S_1, \ldots, S_r)$ of states, where each S_i is from the state space $\mathscr{S}_i$. The (s,t) element in the matrix P_C^i gives the probability that the sth combination of the states of C_0 transits to the tth state in the state space $\mathscr{S}_i$. Using these transition matrices and based on the current states of the character C_0, we can derive the states of C_0 in the next statement. However, note that again the next states are determined randomly.

The impact matrix can be constructed similarly when we have more than one state space, but each state space has an impact matrix. On the other hand, the sociability vector for each character C_0 becomes longer, and a state in a particular state space now may depend not only on the states of other characters in the same state space, but also on the states of other characters in other state spaces. Nevertheless, all these can be easily achieved by simple modification of the model that is based on a single state space.

3.2 Creating Expressions

States, by themselves, are not visible. There is a need to convert the state of a character to a visual image. Visual images will be called expressions of state. We only consider expressions of emotional states since the number of emotional states can be very limited. Another advantage of emotional states is that there are more than one expression for any given emotional state. For instance, a sad person can cry or can have a sad face, a social smile or a neutral face. On the other hand, expressions of physical states has to be unique, i.e. walking can only be expressed by walking. A walking character cannot sit.

Figure 2: A set of expression images for a sea lion, by Ergun Akleman.

Although there is no agreement among the researchers about what the basic emotions are [14], all the

basic emotions suggested by researchers are very limited. They do not exceed twenty. These are anger, disgust, fear, hate, happiness, hope, grief, guilt, interest, joy, love, pain, pleasure, sadness, shame, surprise, rage, terror, wonder, sorrow [5, 4]. To express these 20 emotions, we use 10-14 expressions for each character. An example set is shown in Figure 2. As in this figure, some of the expression images are created by artists directly in a painting program. Other sets of images are created using photographs of real people. Some photographs are further manipulated to achieve more expressive results. As a result, we currently have an approximately more than 50 characters with at least 10 expressions.

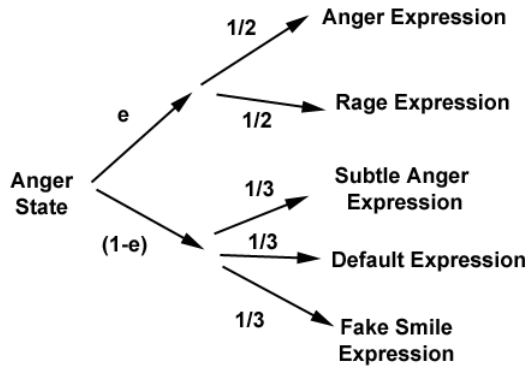

Figure 3: An example of an expression tree.

There is usually no one-to-one correspondence between states and expressions. For instance, an angry person may have a neutral face or a fake/social smile to hide the anger. On the other hand, another person may show the anger. To simulate this variety in people, we introduce another variable called *expressiveness*. For each state, we define a set of expressions for non-expressive characters and another set of expressions for expressive characters. For instance, let us assume that the anger state can have non-expressive expressions Subtle Anger, Default and Fake Smile, and expressive expressions Anger and Rage. Based on this information, we create a tree structure where the probability of each expression is given as the multiplication of the probabilities given each branch. Consider the expression tree shown in Figure 3. In this example, if $e = 1$, with 1/2 probability we choose either anger or rage expressions. If $e = 0$, with 1/3 probability we choose either subtle anger, default or fake smile expressions.

4 Implementation, Conclusion and Future Work

We have developed a prototype system as a proof of the concept. The system is implemented using C# and Microsoft .net framework. The reason to choose this technology is because it has made it easier to build the system with rich functions rapidly. The system includes two main components: a narrative engine, a comic based user interface that is conceptually similar to comic chat [10].

We asked a wide variety of people to create stories using the system. The regardless of the results, we observed that both children and adults like to use the system. The quality of the results depends on the ability of writers and the judgment of the quality, of course, subjective. The system can be used to create short stories for children to teach human interactions. It is also useful to quickly develop some simple story ideas. An example that is created by a bank officer to show how to interact and convince a customer is shown in Figure 4. We observed that this approach can also be used beyond the original intentions of the paper. The system motivates people to write about their experiences and problems and therefore it can probably be used in psychological evaluation similar to Rorschach inkblot Test.

We are thankful to anonymous reviewers, whose insightful reviews that helped us to improve the paper significantly. This work partially supported by the National Science Foundation under Grant No. NSF-CCF-0917288.

References

[1] Roland Barthes. Introduction a l'analyse structurale des racits. *Communications*, 8:1–27, 1966.

[2] Claude Bremond. *Logique du Racit*. Seuil, 1973.

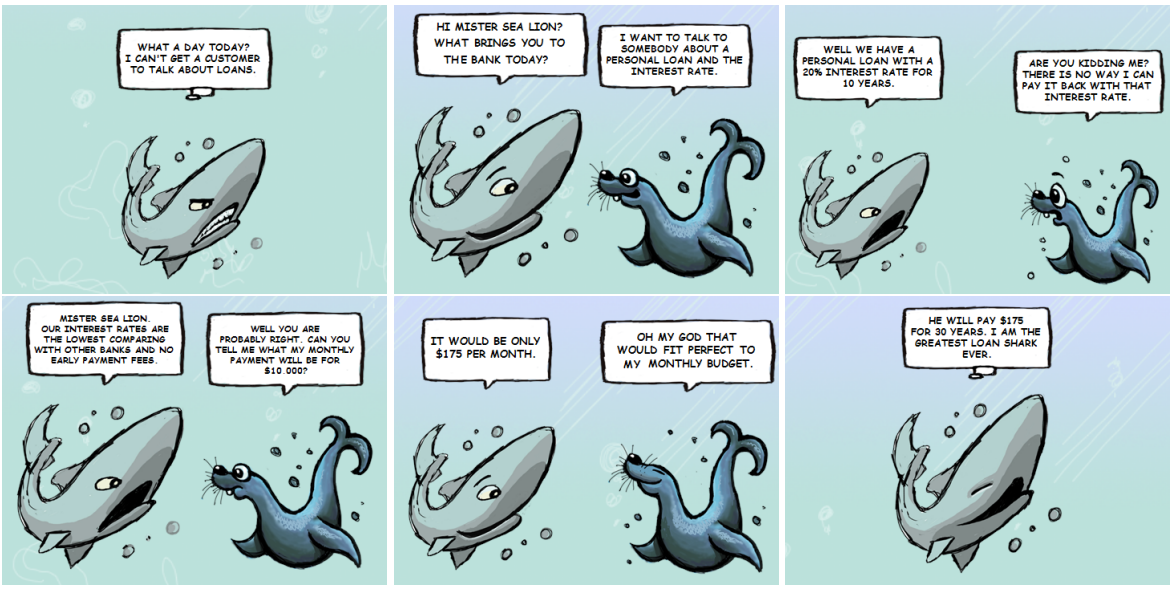

Figure 4: "A day in the life of Loan Shark." A short story piece created using our narration modeler.

[3] Seymour Chatman. *Story and Discourse: Narrative Structure in Fiction and Film*. Ithaca : Cornell University Press, 1978.

[4] P. Ekman. Are there basic emotions? *Psychological Review*, 99:550–553, 1992.

[5] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. *Consulting Psychologists Press*, 1978.

[6] Edward Morgan Forster. *Aspects of the Novel*. Harcourt Inc., 1927.

[7] Algirdas Julien Greimas. *Structural Semantics: An Attempt at a Method*. University of Nebraska Press, Lincoln, Neb., 1983.

[8] Ido A. Iurgel, Nelson Zagalo, and Paolo Petta. The iris network of excellence: Future directions in interactive storytelling, http://iris.scm.tees.ac.uk/. *Proceedings of Second Joint International Conference on Interactive Digital Storytelling, Lecture Notes In Computer Science, Volume 5915*, 2009.

[9] Barbara Korte. Tiefen- und oberflchenstrukturen in der narrativik. In *Literatur in Wissenschaft und Unterricht (18)*, pages 331–352, 1985.

[10] David Kurlander, Tim Skelly, and David Salesin. Comic chat. In *Proceedings of SIGGRAPH 1996*, Computer Graphics Proceedings, Annual Conference Series, pages 225–236. ACM, ACM Press / ACM SIGGRAPH, 1996.

[11] Michael Mateas and Andrew Stern. Facade. http://www.interactivestory.net/, 2005.

[12] Mark Stephen Meadows. *Pause & Effect: The Art of Interactive Narrative*. Pearson Education, 2002.

[13] James R. Meehan. Tale-spin, an interactive program that writes stories. 1977.

[14] A. Ortony and T. J. Turner. What's basic about basic emotions? *Psychological Review*, 97:315–331, 1990.

[15] Vladimir Propp. *Morphology of the Folk Tale, Original work in Russian is published at 1928*. The American Folklore Society and Indiana University, 1968.

[16] Ronald B. Tobias. *20 Master Plots (And How to Build Them)*. Writer's Digest Books, 1993.

[17] Xtranormal. Web-based interactive story construction. http://www.xtranormal.com, 2012.