

To be or not to be: Towards Stateless Interactive Drama

Nicolas Szilas¹, Monica Axelrad¹,

¹ TECFA, FPSE, University of Geneva, CH 1211 Genève 4, Switzerland
{Nicolas.Szilas, Monica.Axelrad}@unige.ch

Abstract. In this paper we investigate a new way to represent the narrative structure in interactive drama models. It consists in explicitly defining generic links between actions, rather than relying on states, as it is the case for other interactive narrative approaches. This new approach is especially promising in terms of authoring.

Keywords: Interactive drama, interactive narrative, authoring, stateless.

1. Introduction

1.1 Interactive Drama systems as state-based engines

Within the field of Interactive Digital Storytelling, Interactive Drama is a computer-based fiction where a user chooses most of the actions for the main character in a story. This is a hard challenge, because it involves both the dynamic generation of narrative events and the integration of user inputs within the generation. In interactive dramas such as *Façade* [11,12], *The Mutiny* (written with IDtension) [24] or *The Balance of Power* (written with Storytron) [22], the user can not only wander in space, use objects and choose among pre-written sentences, as in commercial adventure games, but is also given the possibility of interacting with other characters in the story. The user can, for example, encourage other characters to perform actions, help them, accept or refuse aid, condemn their actions, inform others about past or possible actions, etc.

These three examples, as well as other prototypes that have been developed so far [5,17,19,28], use very different algorithms to generate story events and integrate user's choices. There exists no consensus on how the algorithms of these systems should be classified: plot-based vs character-based, planning vs modular [10], coarse grain vs fine grain [24]. However it is usually admitted that the more generative the algorithm, the more potential for user influence on the story, but the more difficult it is to create an interesting narrative. For example, the plot point approach [20,27], which consists in setting a small set of partially ordered important events in a story, guarantees a good overall quality of the story, but leaves limited room for user

influence (or global agency [11]). Façade is an interesting case, in which generativity stopped at the level of sentences; these are not generated in order to preserve their expressive quality. More generative systems, such as DEFACTO [19], Storytron [22], IDtension [24], EmoEmma [5], Mimesis [28] or Gadin [2] require a more abstract representation of characters' actions, which tend to diminish the expressivity of utterances.

This paper will focus on the latter approach, that is, generative algorithms. Within this category, several approaches can be distinguished:

- Character-based planning [1,4]: consists in calculating all possible story paths from an initial situation and selecting a plan that can reach each character's goal.
- Narrative planning [2,6,28]: consists in setting a story goal as well as various narrative constraints, then calculating a plan that reaches that goal, possibly modified by user's action.
- Narrative actions and constraints [19,22,24]: consists in choosing a parametrized representation of actions and calculating the next action according to narrative constraints.

This short review does not pretend to be a definitive classification of systems but allows us to observe the following: each of these generative systems, while having different formalisms, use generative rules that have states as preconditions (character state, world state, player model state) and other states as a consequence.

This is not a surprising fact, since Artificial Intelligence, with the development of expert systems, commonly uses facts as state descriptions for a world, which are modified by the procedural part of the systems, the rules. But in this paper, we would like to challenge this idea, and see how generative rules could conversely handle actions. That is, how could a generative rule connect one action to another, without the notion of state.

1.2 States vs Actions in Narrative Theories

By analyzing the basic elements of narrative, narrative theorists established some classification of the main micro-constituents of a narrative. In [9], several such theories are overviewed. A basic distinction emerges from this overview, the distinction between static parts (states or stative events) and active parts (non stative events or events). This corresponds to the classical distinction between description and action. Events are then further distinguished between deliberate events (events created by characters, that we can consider actions) and non-deliberate events (also called happenings) and between bounded and unbounded events relative to time. Various narrative genres would use different proportions of these events and states. Typically, a psychological novel such as *Madame Bovary* uses more states than an epic novel. Interactive Drama, being drama, focuses on actions. Thus we are mostly interested in action, in terms of what is displayed to the user, and which choices is given to him.

States however, even if they are not explicitly represented in the story, are usually used to describe the story logic. At a general level, a story is often defined as the transformation from one state to another. In a detailed analysis, Bremond describes

processes, which are a series of local states between which actions can occur [3]. More audience-oriented theories also use states, such as emotional states of the audience [26].

Despite the existence of these narrative theories based on states, we claim that states in narrative, and especially in drama, are *indirect variables*. They are often not directly seen because actions are seen. In fact, states are inferred by the reader, through interpretation [9]. For example, in a movie, we would not objectively see the fact that a character is sad, but we would see him crying. We are not saying that states cannot be expressed in a drama (at a minimal level, it can always be expressed by a narrator), but that a state is more interestingly expressed through an action. This corresponds to the classical writing advice: “show do not tell” (e.g. [13]).

Generative rules between actions naturally lead us to the notion of causality. Although narrative causality is defined in [16] as “A relation of cause between (sets of) situations and/or events”, we claim that the most powerful way to describe narrative causality is to establish a link between an event and another event. For example, in the movie *Alien*, we claim that the relation between “Ash let Dallas enter the ship → crew members are attacked” is more interesting than “Dallas is inside the ship → the ship is infected”. This determines the starting point of the narrative engine that will be described in the rest of the paper.

2. Principle of stateless narrative engine

A stateless narrative engine only handles actions.

Let us take the movie “It’s a wonderful life” as an example. The overarching story can be summarized as follows:

S1: George’s company is ruined

S2: George wants to commit suicide

A0: Clarence shows him the world as it would be if he hadn’t been born

S3: George is horrified and wants to live again

While this first description is correct and conform with the idea of a story as a transformation, an alternative way to describe the same story is:

A1: Uncle Billy tells George he lost George’s company’s money

A2: George starts to commit suicide

A3: Clarence shows him how wonderful life is

A4: George asks Clarence to live again

While the first description contained four states and one action, this second description is purely based on actions. The stories are identical, and a complete description could also be built by interleaving the states of the first story inside the second story, but the second story manages to totally avoid the presence of states. A stateless narrative engine would only contain rules that link different actions together, rather than a rule from a state to an action.

An English description of such a rule could be: “If someone learns he has lost all his money, then he can kill himself”, rather than “if someone is ruined, he can kill himself”.

This exercise of avoiding all states in story description and rules is uneasy. For

example, we are tempted to write that, “Georges is ruined” and later that “George is happy”, action-based analysis can be less natural. Sometimes even, stateless analysis can be more complicated as will be detailed below. It does however open the way to a new, maybe more expressive approach for story representation, dynamic generation and interaction.

3. Rules as links

Right from the start of designing a new narrative engine, we have taken an authoring point of view. Authoring is one of the main issue in Interactive Drama [12,21,23], since the complexity of current approaches and tools has prevented authors from writing compelling stories.

We have attempted to describe the storyworld, the rules between actions, in a compelling way. We have come up with a visual representation of both rules and narrative simulations (stories-so-far), in terms of links between actions.

3.1 Enabling links

An example of a rule is:

$\text{perform}(X,t) \rightarrow \text{informPerformance}(X,Y,t)$

where X and Y are characters and t is a task or an action (a task is an atomic action, author defined, possibly with parameters)

This rule expresses an *enabling link* between two generic actions. The first action represents an action that has already been executed. It is part of the *history* (or story so far). The second action is a possible action. If two actions in the history must be present to enable one possible action, two links are created.

A scenario contains a set of such links, which are activated according to the following algorithm:

Let $E=\{ri\}$ be the set of enabling rules, S the initial set containing the actions already executed (we call it the backstory), P the set of possible actions at a given moment in the narrative. The algorithm is:

BEGIN ALGO 1

 REPEAT

 Fire all rules in E against S , store the right hand side of the firing rule into P

 Remove from P actions whose one antecedent via rules in E are not in S .

 Randomly select an action a in P

 Play a

 Add a to S

 Clear P

 END REPEAT

END ALGO 1

In ALGO1, the selection between possible actions is random. For the player’s character’s actions, the choice can be given to the player. For other actions, the action selection is tackled by another module and algorithm.

When a series of actions has been chosen, the story-so-far can be represented by a simulation graph, which displays actions (mainly their structural description), with their dependence from previous actions. For example, the story sketched in the previous section is represented by a graph of actions with links (Fig. 1). This representation resembles a narrative plan in [18] or causal network in [17], but does not contain any state descriptions.

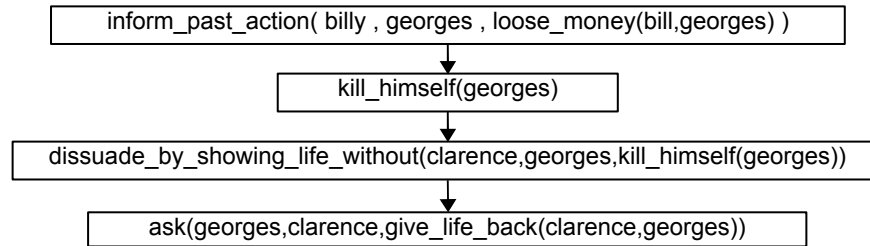


Fig. 1. Graph of actions of a specific story.

In the simplistic case of Fig. 1, the graph is a line, but a single action can depend from several previously executed actions. In the general case, the simulation graph is an oriented acyclic graph.

In the simulation graph depicted in Fig. 1, it should be noted that the link from the first to the second action is different than the link from the second to the third action. The latter one does not wait for the action to be finished to be triggered. Thus, two types of enabling links should be distinguished: *enable when finished* and *enable when started*.

In fact, it is interesting to use this story representation to extend the notion of links beyond simple enabling relations. As such, the model aims to go beyond the classical use of planning operators in story modeling since these operators correspond to enabling links only (irrespective of the fact that they have states in preconditions). Before discussing other types of links, one needs to tackle the issue of nonmonotony, for enabling links.

3.2 Dealing with nonmonotony

Nonmonotony is the property of logical systems according to which what is true at time t might not be true at time $t+1$. This is obviously the case in stories. For example, reasoning with states, if John gives a ring to Mary, Mary has the ring. But if she later sells the ring, she does not have the ring anymore. If we reason with enabling links between actions, the first giving actions would trigger a possible action such as “Mary sells the ring”. Once the selling action is performed, the enabling link is still present, making it possible to sell the ring infinitely! To avoid this, it is necessary to introduce *inhibitory links*. In our example, an inhibitory link connects a selling action to a selling action, to express the fact that “one cannot sell something that one has already sold”. Similarly, another inhibitory link expresses “one cannot sell something if one has already given it”. For this example, if we denote enabling links by “e” and inhibitory links by “e-”, the list of rules is (with $X \neq Y$ and $Y \neq Z$):

$\text{perform}(X, \text{give}(o, Y)) \xrightarrow{e} \text{perform}(Y, \text{give}(o, Z))$
 $\text{perform}(X, \text{give}(o, Y)) \xrightarrow{e} \text{perform}(Y, \text{sell}(o, Z))$
 $\text{perform}(X, \text{sell}(o, Y)) \xrightarrow{e} \text{perform}(Y, \text{give}(o, Z))$
 $\text{perform}(X, \text{sell}(o, Y)) \xrightarrow{e} \text{perform}(Y, \text{sell}(o, Z))$
 $\text{perform}(X, \text{give}(o, Y)) \xrightarrow{e-} \text{perform}(X, \text{give}(o, Y))$
 $\text{perform}(X, \text{give}(o, Y)) \xrightarrow{e-} \text{perform}(X, \text{sell}(o, Y))$
 $\text{perform}(X, \text{sell}(o, Y)) \xrightarrow{e-} \text{perform}(X, \text{give}(o, Y))$
 $\text{perform}(X, \text{sell}(o, Y)) \xrightarrow{e-} \text{perform}(X, \text{sell}(o, Y))$

When an action is both triggered by a positive enabling link and a negative one, the most recent is winning. Thus, the new algorithm is, with E^- denoting the set of inhibitory rules and P^- denoting the set of impossible actions:

BEGIN ALGO2

REPEAT

Fire all rules in E^- against S , store the right hand side of the firing rule into P^-

Fire all rules in E against S , store the right hand side of the firing rule into P

Remove from P actions whose one antecedent via rules in E are not in S .

Remove from P all actions for which there exist a newer version in P^- ,

Randomly select an action a in P

Play a

Add a to S

Clear P and P^-

END REPEAT

END ALGO2

This example also illustrates an issue with the stateless approach: more rules are necessary to describe the story logic.

3.3 Relevance links

Relevance links are inspired from the maxim of relevance described by Grice, in dialogic interaction. This maxim says that people act according to what has been acted before, they follow the current topic of conversation. The relevance link is thus particularly useful for dialogic actions (assuming the hypothesis that dialogues are also managed by the narrative engine, rather than a sub-engine).

The problem of action relevance is typical in Interactive Drama. In [27], it was described as *thought flow*, which consists in favoring actions that naturally follow the one played just before. In IDtension, a mechanism of relevance was also implemented, to manage dialogues [24]. Relevance links are also related to some research on dialogue games between agents [15].

Relevance links work differently from enabling links. They do not make an action possible or impossible, but rather favor one action instead of another. They are used in the algorithm to weight and rank actions already selected according to enabling and inhibitory links. The weighting factor is calculated as follows:

Let us denote a the possible action to be weighted, $Wr(a)$ the corresponding weight, $R(a)$ the set of executed actions for which there exists a relevance link to a , $time(x)$ the time at which the action x has occurred or will occur:

$$Wr(a) = 1 + \sum_{p \in R(a)} \sigma(\text{time}(a) - \text{time}(p))$$

where σ is a thresholding function, that is maximal for small values and null for larger ones. For example, if the time is measured in a discrete manner (one played action is a time tick), σ could be defined so as to $\sigma(1) = 1$ and $\sigma(3) = 0$, to express the fact that the action is not relevant anymore when two or more actions have been interleaved between the past action and the current one.

3.4 Explanation links

The links between two actions is sometimes weaker than an enabling link. Let us take the example of revenge. If somebody's loved one is the victim of a degradation action, then revenge, for this character, consists in repeating the action, but against the aggressor. The action of revenge must already be possible, but becomes narratively interesting because of the initial action. This kind of link is called an explanatory link, because the resulting action is explained by the initiating action.

As with relevance links, explanatory links will not enable or disable some actions but reinforce an action over another. Let us denote $Wx(a)$ the weight of action a regarding explanation, $X(a)$ the set of executed actions for which there exist an explanation link to a :

$$Wx(a) = 1 + \sum_{p \in X(a)} l_x(p, a)$$

where $l_x(p, a)$ is the strength of the explanatory link between p and a . We have introduced here the idea that a link can vary in strength, defined by an author. For example, a given link involving revenge might have a higher strength than another link involving gratitude.

3.5 Link structures

The three types of links that have been presented above constitute a basis for building an interactive narrative. With this basis, many temporal simulations can be created which are not stories (the system over-generates). The way the author will organize the generic links between actions will be essential for effectively obtaining a narrative experience. Links can be used at both local and global levels. If a character gives an object to another one, it can be motivated by the simple fact that it is possible to do so because he was given the object previously. But it can be also motivated by the fact that this character failed previously in a task necessitating two persons. This latter case corresponds to a sort of higher level goal for the character, although the goal itself has been shortcut by the stateless approach.

Non formally, interesting link structures are those which combine low level links with higher level links, with long term dependencies. Interesting link structures are not linear as in Figure 1 but contain many incoming links from various types and time distance.

4. Preliminary results

In order to evaluate the feasibility of the stateless narrative engine, we have developed a simulation with a few basic rules that generate story actions over time. The idea is not to demonstrate the quality of the generated story, which is naturally bound by the minimal number of rules that are currently implemented and the lack of a proper action selection mechanism. Rather, our goal is to illustrate how the proposed approach can be programmed, and open a discussion on its potential and limitations.

The minimal scenario on which our narrative simulation operates consists of some children who give various objects to each other, and sometimes thank each other for the presents. From an initial condition, the narrative simulation produces an arbitrary number of successive actions. User action is not considered here, but could be easily added, by choosing one character as the player character and letting the user choose among all possible actions for this character.

The story world contains four characters (Allan, Bill, Chloe and Ed) and three objects (the ball, the stick and the box). Two tasks are possible: $give(x,y,c)$ (“x gives to y an object c”), $thank(x,y,t)$ (“x thanks y for having performed t”). Four rules are programmed:

$give(x,y,c) \text{ — } e \text{ — } \rightarrow give(y,z,c)$: one can give an object that one was given
 $give(x,y,c) \text{ — } e- \text{ — } \rightarrow give(x,z,c)$: one cannot give an object that one has given
 $give(x,y,c) \text{ — } r \text{ — } \rightarrow thank(y,x,give(x,y,c))$: one should thank for an object
 $give(x,y,c) \text{ — } x \text{ — } \rightarrow give(y,x,c')$: one could give an object to thank for being given another object

To initiate the narrative simulation, a set of actions already executed before the start of the story is added to the history (the back story):

$give(\text{someone}, \text{allan}, \text{ball})$
 $give(\text{someone}, \text{bill}, \text{stick})$
 $give(\text{someone}, \text{chloe}, \text{box})$

Note that “someone” is in fact a fifth character, a special character that is not involved in any action in the played story.

The example was programmed entirely with JESS [8], a rule-based system integrated with Java. The simulation result is reproduced in Table 1. It shows 10 actions generated by the story. Text describing actions is automatically generated in a basic form. This simple simulation, while far from representing a full story, illustrates the circulation of various objects between characters. The characters give objects to each other in a coherent manner, while thank actions systematically follow giving actions, according to the relevance link. Note that there is no distinction here between the “give” produced as a thanking of a previous give, and a “normal give”. Indeed, we want this distinction to appear at the level of text generation, according to the various links incoming to the played action.

Table 1. Elementary narrative simulation of a storyworld without states.

Bill gives Chloe the stick
Chloe thanks Bill for the stick
Chloe gives Bill the box because Bill was generous last time
Bill thanks Chloe for the box
Allan gives Bill the ball
Bill thanks Allan for the ball
Bill gives Chloe the box because Chloe was generous last time
Chloe thanks Bill for the box
Chloe gives Bill the stick because Bill was generous last time
Bill thanks Chloe for the stick

5. Discussion

We have shown an example of an elementary story simulation based only on the notion of links as the generic relation between two actions. In other words, no state is stored in the world of the story, only past actions are stored. Computationally speaking, this has no specific advantage, but we believe that this alternative representation of the story dynamics has potential benefits in terms of both authoring and expressiveness.

As explained in the introduction, existing interactive drama engines have extensively used the notion of states to describe how events are generated. While this notion is very natural for Artificial Intelligence researchers, it is far from being intuitive for authors. It is more natural to say that: “Harry decided to jump because his friends have encouraged him to three times” rather than “Harry decided to jump because his level of motivation was successively increased by the encouragement of his friends, and reached a certain threshold”. It could be argued however that the character-centered design that is often adopted by writers or screenwriters [7] consists in defining a rich set of traits and psychological states and moods. Similarly, a screenwriter might think in terms of dramatic tension, a narrative state that is either incremented or decremented by narrative events. So it would be wrong to claim that states are not part of an author’s way of thinking. But it is doubtful that authors are willing to push the notion of state up to an algorithmic level. Therefore, we claim that a stateless narrative engine is an interesting alternative, possibly closer to dramatic thinking. In the field of interactive digital storytelling, Storytron [22] is a pioneer Interactive Drama system that mixes the two approaches. In Storytron, characters are heavily defined by variables (21 personality traits) but actions (called verbs) are directly linked to each other by the author. It is interesting to report the feedback from one of the authors of this system: “I reached the point where I was juggling so many variables in my decision scripts that they became confusing and downright useless [...] The linkages were less of a problem, though not trivial”. Apparently, linking

actions together was easier (though “not trivial”) than reasoning on characters' states.

Another potential benefit from a stateless narrative engine is debugging. When authoring an interactive drama, it is always difficult to identify what caused one action to trigger instead of another. It usually requires inspecting the internal state of the engine, and multiple variables. In the proposed approach, what needs to be inspected are the various links from past actions to the action in question. We expect such a narrative engine to be more transparent to a non-technical user.

Furthermore, we see great potential in a stateless narrative engine in terms of expressiveness. Every time an action is selected, we directly have access to all previous actions that caused (in the large sense) this action. It opens the way to richer output expression. For example, to express the giving action, one can say “I give you the ball I have” but can also say “I give you the ball that Chloe gave me yesterday” which is not only more complete, but more relevant as well, especially from a narrative point of view.

Finally, links between past actions and future actions could be used for the user interface. In a menu-style interface, it would consist in selecting a past action and choosing which possible action this past action enables or explains or immediately arouses (according to enabling, explaining and relevance links respectively). Such an interface would extend the history-based interface proposed in [25].

These potential benefits of a stateless narrative engine come at a price. As illustrated in Section 3.2, without states, more links must be written. A state, like “having” concentrates all actions whose consequences are to have an object and all actions that result from this possession. Without states, each causing action must be explicitly connected to each consequence action. The number of connections increases not linearly but polynomially (second degree) with the number of actions. To solve this problem, that has not yet arisen given the simplicity of the simulations performed so far, we see two solutions. First one could design an authoring tool that is able to intelligently assist the author, by automatically adding links when a new action is added, according to regular patterns of links that are inferred. Second, the notion of states could be implicitly reintroduced by concentrating a bundle of links into a single edge in a hypergraph, linking several actions to several actions.

A last question that we would like to discuss here concerns the action selection algorithm. In fact a stateless narrative engine is a way to represent actions and their interdependence but does not favor a specific type of algorithm over another. It is possible to use a one step action selection mechanism, as in IDtension [24] or Defacto [19], but planning algorithms can also be used, as in I-Storytelling [4] or Mimesis [18,28]. We foresee a large range of algorithms that could be used on a story representation based on links. By simulating the forward chaining of the links, it becomes possible to plan a story according to a final action or set of actions that must be achieved for the story to end. It is also possible to navigate through the set of played actions. For example, a full history of an action can be made available, if one looks at the action that enabled the action that enabled, ... the current action. This could generate some dialogue lines such as “Here is the ball that Steve robbed you, before giving it to me”.

6. Conclusion

In this article, we presented initial experiments using a stateless narrative engine where actions are generated through author-defined, generic rules between potential actions. The writing and reasoning that goes into the interactive narrative used in this approach is closer to the true essence of narrative compared to state based models. It is still undetermined if this approach can be or should be adopted without compromise, or if at some point, states should be reintroduced into the architecture. In any case, the notion of generic links between actions is an interesting and new approach in the field.

From an algorithmic perspective, the model could be extended into two directions. First, new types of links could be introduced to deal with the subtle relations in narrative, such as repetitive actions, symmetries, or topic-based relations. Second, several links could be combined together within a structure, in order to include more elaborate narrative patterns.

From an authoring point of view, the model will be quickly associated to a simple authoring tool where links can be easily entered and the execution monitored. While narrative theories can certainly provide us the grounds for creating proper link types and organizing them, we think it is more efficient to guide this process of design according to real authoring needs.

References

1. Aylett, R., Louchart, S., Dias, J., Paiva, A., Vala, M., Woods, S.: Unscripted narrative for affectively driven characters. *IEEE Comp. Graphics and Applications*, 26(3), 42-52 (2006)
2. Barber, H., Kudenko, D.: Dynamic generation of dilemma-based interactive narratives. In: *AIIDE'07*, Stanford, California (2007)
3. Bremond, C.: *Logique du récit*. Seuil, Paris (1973)
4. Cavazza, M., Charles, F., Mead, S. J.: Characters in Search of an author: AI-based Virtual Storytelling. In *Proceedings of the First International Conference on Virtual Storytelling - ICVS 2001*. LNCS, vol. 2197, 145-154. Springer, Heidelberg (2001)
5. Charles, F., Pizzi, D. Cavazza, Vogt T., André E. EmoEmma: Emotional Speech Input for Interactive Storytelling (Demo) *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-Demos)*, Budapest, Hungary (2009)
6. Cheong, Y.G., Young, M.R.: Narrative Generation for Suspense: Modeling and Evaluation. In: U. Spierling and N. Szilas (eds.) *Proceedings of the First Joint Conference on Interactive Digital Storytelling (ICIDS)*. LNCS, vol 5334, pp. 147-158. Springer, Heidelberg (2008)
7. Egri, L.: *The Art of Dramatic Writing*. Simon & Schuster, New York London Toronto Sydney Tokyo Singapore (1946)
8. Friedman-Hill, E.: Jess, the Rule Engine for Java. Sandia National Labs. <http://herzberg.ca.sandia.gov/jess/> (1995-)
9. Herman, D.: *Story Logic. Problems and Possibilities of Narrative*. University of Nebraska Press (2002)
10. Magerko, B.: A Comparative Analysis of Story Representations for Interactive Narrative Systems. *3rd Annual Artificial Intelligence for Interactive Digital Entertainment Conference*, 2007. Marina del Rey, California. (2007)

11. Mateas, M., Stern, A.: Structuring content in the Facade interactive drama architecture. In: Artificial Intelligence and Interactive Digital Entertainment Conference - AIIDE 2005 (2005)
12. Mateas, M., Stern, A.: Procedural Authorship: A Case-Study Of the Interactive Drama Façade. Digital Arts and Culture (DAC) (2005)
13. McKee, R.: *Story: Substance, Structure, Style, and the Principles of Screenwriting*. HarperCollins, New York (1997)
14. Mixon, L.J.: I CAN'T BELIEVE I DID THAT: A Reflection on Some of My Biggest Shattertown Blunders. http://www.erasmatazz.com/library/StS_Blunders.html (accessed July 2009)
15. Morge, M. A dialogue game for agent resolving conflicts by verbal means. Proceedings of the second workshop on Logic and Communication in Multi-Agent Systems (LCMAS), Nancy, France, August (2004)
16. Prince, Gerald. *A Dictionary of Narratology*. University of Nebraska Press, Lincoln & London (2003)
17. Riedl, M.O.: *Narrative Generation: Balancing Plot and Character*. PhD thesis., North Carolina State University (2004).
18. Riedl, M.O., Saretto, C.J., Young, R.M.: Managing interaction between users and agents in a multiagent storytelling environment", Second ACM Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2003, pp. 741-748. ACM Press (2003)
19. Sgouros, N. M.: Dynamic, User-Centered Resolution in Interactive Stories. In Pollack, M. (ed.) *IJCAI'97 Proceedings of the 15th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, San Francisco (1997)
20. Skorupski, J., Jayapalan, L., Marquez, S., Mateas, M.: Wide Ruled: A Friendly Interface to Author-Goal Based Story Generation. In Proc. of the Fourth International Conference on Virtual Storytelling - ICVS 2007, LNCS, vol. 4871, pp. 26-37. Springer, Heidelberg (2007)
21. Spierling, U.: Adding Aspects of "Implicit Creation" to the Authoring Process in Interactive Storytelling. In Proceedings of the Fourth International Conference on Virtual Storytelling - ICVS 2007, LNCS, vol. 4871, pp. 13-25. Springer, Heidelberg (2007)
22. Storytron, <http://www.storytron.com>
23. Szilas, N., Marty, O., and Rety, J.-H.: Authoring Highly Generative Interactive Drama. In Proc. of the Second International Conference on Virtual Storytelling - ICVS 2003 (Toulouse, France), LNCS, vol. 2897, pp. 37-46. Springer, Heidelberg (2003)
24. Szilas, N.: A Computational Model of an Intelligent Narrator for Interactive Narratives. *Applied Artificial Intelligence*, 21(8), 753-801 (2007)
25. Szilas, N., Kavakli, M.: PastMaster@Storytelling: A Controlled Interface for Interactive Drama, *IUI 2006: International Conference on Intelligent user Interfaces*, Sydney, Australia, 29 January to 1 February 2006, pp. 288-290 (2006)
26. Tan, E.: *Emotion and the structure of narrative film. Film as an emotion machine*. Erlbaum, Mahwah (NJ) (1996)
27. Weyhrauch, P. *Guiding Interactive Drama*. Ph.D. Dissertation, Tech report CMUCS-97-109, Carnegie Mellon University (1997)
28. Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R.J. Saretto, C. J.: An architecture for integrating plan-based behavior generation with interactive game environments, *Journal of Game Development*, 1, 1 (2004)

Acknowledgements

This work has been funded (in part) by the European Commission under grant agreement IRIS (FP7-ICT-231824).