An approach to Collaborative Storytelling

Guilherme António Mesquita Barbas Fernandes
guilherme.fernandes@ist.utl.pt

Instituto Superior Técnico, MEIC Taguspark
Grupo de Agentes Inteligentes e Personagens Sintéticas

Abstract. Interactive Storytelling has the potential to greatly benefit the entertainment industry and as such it has been the subject of many academic research in recent time. However, creating content to such systems still proves to be a very difficult task for various reasons such as the need to anticipate the choices taken by the user. To try and solve some of the problems related to the creation of content for such systems we use the concept of Group Authoring. By bringing more than one author into the picture, the level of choices and their quality can be increased.

Keywords: Interactive Storytelling, Authoring, Group Authoring, Collaboration

1 Introduction

Given its enormous potential Interactive Storytelling(IS for short) has been the subject of many academic research in recent time. It takes place between the user and the computer and its precisely that symbiotic relationship, that is built by both, that makes IS so interesting and with so much potential. Whether it is books [17], films [5] or games [3], the connection that is established and the way IS changes the potential of the medium are all exciting and extremely promising.

When examining in greater detail this area a question presented itself:"Why don't we see this type of stories more often?" that's when the Authoring Challenge in IS [12] became apparent. The main limitation found nowadays in the general acceptance of this kind of stories has been not in the technological department but more in the way of how do we create this type of stories. In order to solve this problem it was decided to try and use the benefits and advantages of shifting the creation process from a one author situation to a group. However, some questions arose "How do people tell/create stories in a group/collaboratively?", "Can we use that mechanism and apply it to IS?" among others. In an attempt to find an answer to this questions existing activities that already used the group element as an advantage rather than an hindrance were explored. It was during that research that it became noticeable the existing *pen-and-paper* RPGs(Role-Playing-Games) where some parallelisms can be drawn with IS (the existence of a Game Manager/ Drama Manager is a good example of that) and inclusively some of this RPGs already have the objective of creating a story collaboratively; for the purpose of this work we focused mainly on *Universalis* [13].

2 Concepts and Applications

Let's start by answering an important question: *What is Interactive Storytelling?* Although this term is loosely used, the truth is it was not until [7] that the now accepted definition came to be. Crawford defined Interactive Storytelling as a form of digital entertainment in which the user takes the part of a lead character in a narrative experience. This mean the user *"incarnates"* a certain character in a fictional world being given the power to take the decisions inherent to the character. This means, for instance, in Star Wars[1] it's up to the user to decide if Luke should finish his Jedi training in Dagobah to save the Universe or head to the Cloud City to save his friends.

[1] Star Wars Episode V: The Empire Strikes Back, *Lucasfilm*, 1980

2.1 Strategies

Branching Trees The first efforts on trying to implement an IS system used this as the initial strategy. It consists on predicting what types of choices the player will want to take and organize the entire story around them [20], creating a state for each choice and making the player navigate through those same states. Then youmust be wondering *"Why isn't this strategy used more often?"*. Well, let's suppose that the player is only offered 2 choices at each level. With 10 levels, the author will have to create $2^{10} = 1024$ different states. Add to that the amount of players that only play once through a story-based game and it becomes clear why this method is not feasible.

Foldback Schemes Foldback Schemes are very similar to the branching trees with one main difference. Instead of each choice generating a completely different state, there are key points where the story must pass and although some liberty is given to the player, those points are always present in the story [1]. An example of this strategy is represented in the figure 1.

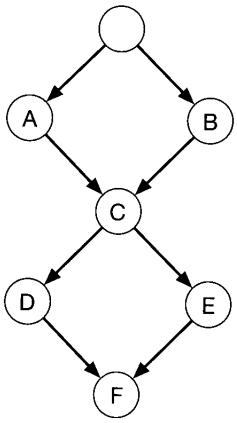

Fig. 1. Example of a Foldback Strategy

2.2 Drama Managers

Stories are all about passing knowledge and/or information. In an Interactive Story, that knowledge is passed through the interaction with the world, whether it is by witnessing an event, talking to a Non-Player Character (NPC), etc. Under these assumptions it is normal to think *"why don't we delegate the responsibility of creating the story to the NPCs?"*.

Well, the fact is that although NPCs do have interference with the story in the sense that they influence the choices and actions of the player, they don't share a common 'mind' and therefore lack the ability to direct the story in the path intended by the author. To circumvent that a central mechanism guide the "Drama" is necessary. This Drama Manager (DM for short) notion relates directly to the *Director* described by Laurel[11] and Bates [2], something to "Dot the i's and cross the t's" and keep things going smoothly for both the author and the player.

2.3 Authoring

Authoring raises several problems and presents itself as being one of the major problems to Interactive Storytelling. Not only the process to create an interactive story is extremely different than

creating a linear story in the lines of *Little Red Riding Hood* [23]. The creator must also imagine the setting where the action is going to take place. For last, after analysing each type of these systems to control the Drama it becomes clear that to achieve IS the story must be put in terms that the DM must understand. This however, is not very straightforward for writers whose main focus is centred through characters' dialogue. This arises from the very simple fact that, the majority of authoring systems are created for already existing IS architectures and usually they derive themselves from debugging tools for those same architectures, as observed in [18].

Authoring Requirements Authoring for IS is extremely different than authoring for classic Storytelling. Timelines, relationships, the Drama Manager permitted actions and required representation all become necessary in an IS system. That's why some work has been done to properly identify the requirements for a generic authoring system in IS. In particular, Magerko, using his Scribe [14] system, identified what he considers six crucial points that every authoring tool must adhere to. *Generality*, the system must be general enough not to limit the creativity of the author, it must not focus itself solely on a determined environment or story setting, opposed to what happens with most authoring tools available for the general public, as is the case of [6]. *Debugging*, due to the complex nature of IS, it's typical for the tool to offer some sort of debugging or else the entire authoring becomes chaotic and at some times it could become random as the author has no idea if what he is creating works or not. *Usability*, is a sub-area of Human-Computer Interaction (HCI) study [8] [19] in a field of expertise far greater than IS, this requirement states that a program must be easy to use and learn. *Environment Representation* denotes the space where the action is taking place, the physical location with its entire set of rules and limitations. *Pacing and Timing* are a concern in linear storytelling and in IS they gain a new dimension, since actions and events may be preceded by other. *Pacing and Timing* are a very important part of IS [9] and must be dealt with specifically. *Scope* is the necessary authoring for the story: character behaviour, dialogue scripts, world restrictions, etc. must be supported by the application.

2.4 Group Authoring

Group Authoring has been a subject of much research in Computer Science. The need for people to work and collaborate together made Group Authoring a necessity instead of simply an academic research field. Group Authoring has its roots set in collaborative work or collaboration. It was Terveen [24] who coined the definition: *"Collaboration is a process in which two or more agents work together to achieve shared goals"*. In many cases this approach proves not only to be better suited for the task at hand, but also proves to be a necessity.

Collaborative Storytelling A specific subset of Group Authoring is the so-called Collaborative, in this method a group of users gather to produce a story as one. Like in Group authoring this method relies in the group as a whole to keep the quality of the story as high as possible, however there is a fundamental difference between both: While in Group Authoring the objective is to create a story to be experienced by other users, in Collaborative Storytelling the experience of creating the story is the main goal.

While in Group Authoring each user creates a different part of the story with a certain degree of autonomy, in Collaborative Storytelling all users are engaged in the same task at all times which creates a certain redundancy which, in theory, makes the same task take longer to be completed.

It should be noted that the work made in this area usually relies on tangible objects to create the story. Some projects regarding collaborative or group storytelling have appeared, such as Zuzie [16], PuzzleTale [22], KidPad [10] and StoryMat [21] who use both tangible objects and the combination between them to create a story. It should be noted that authoring requirements like *Generality* are not supported since the story is highly dependent on the items used and the meaning attributed to them, with very few works done without that limitation.

2.5 RPG

Role-Playing games are a genre of game that usually involve a story where the player takes part as a character or group and the action/story is usually controlled by a GM (Game Master). As it's noticeable there are many similarities between IS and RPGs, one may even argue that in its purest form RPGs are interactive stories with the GM adapting the game to better suit the players needs. Therefore, it comes as no surprise that by analysing some of the concepts and techniques used to improve RPGs we may get ideas and notions that can be used to similarly improve IS.

Universalis One place where many improvements were found was in the *pen-and-paper* RPG *Universalis* [13]. *Universalis* is a Collaborative Storytelling RPG where the objective is not to win but to create a story. Acting more as a framework than a game it presented several characteristics capable of helping in the creation of a Collaborative authoring tool for IS.

Currency One of the very interesting aspects of this game is that in order to maintain balance between the amount of influence each player can exert in the story it effectively creates an objective and quantitative method to do so. Each player gets assigned an X number of coins in each turn and these coins represent what the player who possesses them can do. Every action in *Universalis* to be considered a "fact", something that for the effects of story really happened, needs at least a coin to be spent in it.

Story Components In *Universalis* each component of the story whether it is a character, an object (within the *Universalis* universe this is called a Prop) or a location must be described and defined. Each story element is defined by the following characteristics: *Role*, *Attributes*, *Possessions* and *Name*.

Creation When creating a new Story Element the first thing we should define varies according to the type of Story Element, if it is a location the first thing we establish is its name, otherwise the first thing we define is the elements' role. It should be noted that although possible it is not necessary to give a name to neither a character nor a prop, for instance if Patu was to fight an enemy soldier, it is not necessary to define that soldier's name. It can simply be identified by its role. As it has been stated the creation of an element requires the spending of coins in order to be definitive.

Scenes Each story in a session of *Universalis* is divided into Scenes. Each scene is characterized primarily by:

- Time - the *when*
- Place - the *where*
- Events - the *what*

Each scene also contains actors, elements which interact within the scene and are the targets(Characters or Props) of the events that occur. With all these aspects combined together we are able to create an entire Scene with its own story, set on its own time which happens in its own place.

Fines and Complications In order to maintain the rules of the game, it is possible to penalize each player in a number of coins every time they break a rule, for instance, if it is defined that no references to Transformers can be made. If a player breaks this rule he can/should be punished.

3 Architecture

Romans had a saying: *Divide et impera* which translates to "Divide and Conquer", to achieve our goals that's the strategy we used. Since the problems we are facing are far too complex and ambitious to just start testing/solving we need to divide them into smaller ones to be able to see the "big picture" and in that way to delineate a feasible solution. The following sub-problems were the ones we considered critical:

1. Define a model capable of representing Interactive stories
2. Define a method to create Interactive stories in a collaborative context
3. Create a prototype capable of creating Interactive stories in the model defined in 1 using the method created in 2, independent from any DM, capable of being used by authors with no previous programming experience.

In the next sections we are going to be analysing each of this problems individually and in greater detail.

3.1 Story Model

First of all, let's start by remembering that in *Universalis* a story is a group of Scenes, it is not strange then that we used that as a foundation for our story. In *Universalis* a scene is composed by: a location, a set of characters, a set of props and a plot (the other aspects considered like Social Contracts and Rules Gimmick only regard the method to create the story) in terms of a linear story these points are enough to define an entire script as such they were directly ported to our solution. However, when the aspect of interaction is brought to the table, these features, although still necessary, become insufficient. Some adaptations needed to be made in order to be able to represent an Interactive story. In the following sections we are going to mention some of the changes needed to be made and also describe our Story Model in a greater detail.

Scene Description In *Universalis* the Scenes are the foundation, the building blocks if you will of the story. A Scene is characterized by three main aspects: the Location **where** it happens, the Time **when** it happens and **what** happens in that Scene (the Plot). We don't consider characters and props on the three main aspects because they can be slotted inside the Plot. By following this logic/model we can clearly see that Characters and Props don't exist outside a Scene, they only exist in a moment in time within that scene.

Time *"Order is a feature of all narratives"* [15], as such our end result will also contain within itself an order, not just one by that matter but several, since in Interactive Stories the order can be permuted. In *Universalis* time is defined only by a single sentence, for instance "forty years ago", "last night", etc. Its meaning is entirely semantic. Although that form of identification was incorporated into the prototype it proved to be insufficient as it didn't allow for multiple paths and different decisions. In order to support Interactive Stories the Foldback Schemes strategy [1] was implemented to define the actual order between scenes.

Plot Current methods used to obtain Interactive Storytelling all rely on the use of the so-called Drama Manager. As it was also stated it was necessary for the DM to know which actions are supposed to be happening, to facilitate this job and to easily identify what actions the DM must focus on we have separated from the usual narrative the *events*, to which we called Actions. This not only allows for a greater control of what happens on a scene but it clearly establishes a separation between important facts and events that only contribute to enrich the scene (what is commonly called color).

3.2 Authoring Method

Once solved the problem of how to represent an interactive story was solved our focus shifted to the process regarding the creation of such stories. Our main idea was, similarly to what happened with the domain, to start from the *Universalis* method and from there adapt it to fit our needs. In order to keep the Rules separated from the story this module is independent from the Domain and all the alterations to the Domain must pass first through here, furthermore this module is responsible entirely for the author mechanics, i.e. who has control of the scene, how many coins should be refreshed, managing complications, etc. The process in itself remained almost unchanged with the only alterations being the way complications develop, and the ability to change past scenes; the main reason not much has changed from the original version is due to our tool also being used for collaborative creation and to be used in turns which makes the process very similar.

Complications Before if you didn't want the story to go a certain way a Complication was the only way to prevent this. With our Model this no longer occurs, if a participant does not wishes the story to unfold in the direction it currently is he only needs to take control of the next Scene and create a different path. To accommodate for its new functions we slightly adapted the mechanism, in the original version a Complication could have several sides with, for instance, four different players fighting for four different outcomes. In our version we only allow for two sides to take the "battlefield" since our Complications mechanism only serves as a quality assurance mechanism, it is to be used mainly if the story is going through a really bad path.

Past Scenes When first testing one of the first versions of our application it was noticed that most of the times an error had been made or an event would have been forgotten in previous scenes. To account for the need to sometimes change the past we removed the rule that prevented us from doing that and allow for the players to do as they wished. However, full control without rules would make it possible for players to completely rewrite the past to prevent this from happening by forcing the player to spend all the coins he said he would in the Bidding Phase for the scene he created.

3.3 View Model

The story is in the Domain, the method used are the rules but how all that is presented to the users has not been mentioned yet. In order to better organize our methods it was important to make the presentation part one which was easily edited to test several solutions. To accomplish that it was decided that should be a main module which according to the information received from the rules module changed the presentation state, if we wanted to change the way a method worked or add a different state we simply needed to call it from the rules and it would automatically call what is presented to the user.

3.4 Prototype Architecture

In order to better modulate and operate our system we divided it into three parts: Domain, Rules and View. Heavily based on the MVC [4] software architecture, our model is designed to enable the test of several solutions with ease and to completely separate the story from the method used to create/alter it and to visualize it. In order to achieve that result we can tackle each of the sub-problems individually simply by modifying one of the parts, the story model is contained within the Domain and if we wanted to test any different model we simply needed to change the Domain module. If we intend to alter the balance, the order or basically anything we can do it in the Rules module, again without the need to change the other modules. The View basically relates to the display and input of the user, if we wanted a box of another color, a bigger image, anything graphical basically the View module is the one in charge.

4 Implementation

As previously mentioned the work developed on the field of Authoring in IS is, although blooming, still insufficient for the objectives we propose. The tools used are not easily available and most important they are almost all dependent of the DM system they were created for. For this reasons it was necessary for us to literally start from scratch. It was then necessary to choose a programming language/ framework on which to program our prototype.

The choice made resulted in using Swing [2] to create our prototype (which was named StoryColla). Since the tool-kits and libraries made available by the massive community it has made the prototyping phase extremely fast for simple experiments, also its extensibility and platform independence proved to be up to the task of what we pretended.

4.1 Structure

The prototype is structured in modules each one responsible for managing certain functionalities, to which we called managers. We have three main Managers responsible for the main aspects we identified in our architecture (Domain, Rules, View) and we have several smaller ones responsible for accessing and modifying specific aspects in the Domain. All managers and respective functions are described below:

ViewManager: What the user sees, everything that is displayed is the responsibility of this Manager. Further explained in Section 4.1.

GameManager: Responsible for the process, it acts as an intermediate between the Domain and the visualization, it is where the rules are enforced and where the validity of the domain is maintained. This manager is also important because it is our "Main Manager", it's where the View Manager and the DomainManager are kept.

DomainManager: Where the story is kept, any change to the story must go through here. It is explained in further detail in 4.1.

CharacterManager/PropManager/LocationManager: As the name implies this module was in charge of creating/modifying characters.

AttributeManager: An inherent component of every Character/Prop/Location is the attribute, whether it is the role or simply a characteristic it is necessary to keep a manager coordinating the attributes.

SceneManager: This module is responsible for the story *per se*. The creation/edition of a scene must all be done through this manager.

ImportManager: This manager was added as a result of adding the functionality to import elements from previously created stories, its only functions are to store the imported elements and provide access to them.

Domain Manager This Manager is responsible for keeping the Story coherent and also to serve as a bridge between the other Models and the domain. This Manager has several dependent Managers to better control the underlying story such as the Props, Character and Location Manager. By dividing the functionalities through several managers we are able to maintain all of them simple and focused only on small tasks.

View Manager Until now we only talked about the representation of the story and how to access it, this manager is in charge of other equally important aspect, to show it to the user. By switching between states the ViewManager is capable of showing what we want the users to see and also where they introduce their input, making the alterations made to the interface completely separated from the ones made in the domain.

[2] `http://download.oracle.com/javase/6/docs/technotes/guides/swing/`; accessed 15-June-2011

5 Tests Results and Analysis

Our tests were performed by a total of 17 people, in a total of 5 groups, with ages ranging from 21 to 31 years old of both genders. Since we wanted to test if our system was usable by non-programmers 23% of the participants had no prior experience in computer programming. The tests also attempted to ascertain how well did StoryColla bode against the Authoring Requirements identified in 2.3 and to validate the decisions taken when designing the system.

5.1 Authoring Requirements

Test results have shown that with the exception of Timing and Pacing all Authoring Requirements were fulfilled, it should be noted that generality isn't mentioned here because *Universalis* already accomplishes that requirement out of the box.

Debugging In the StoryColla system, in average, users rated the clarity of the error messages in 3,3 (in a 1-5 range) with an average deviation of 0,8. In terms of usefulness of that information 52% of the users considered the information to be insufficient to identify and recover from the errors that caused them.

Usability As a whole 94% of the users considered the usability of the entire system at least acceptable and not experience damaging. Concerning the users who had no previous programming knowledge all of them considered the prototype usable and found that in no way it damaged their experience significantly, although some problems were identified by them; Such as the extensive need to use the keyboard which all of them with no exception found exhausting.

Environment Representation When asked about the Prototype strong points 47% of all users considered the representation of the environment and characters the aspect that really stood out. In a scale of 1 to 5, the average given by the users to the adequacy of the representation of elements was 3,94 with an average deviation of 0,54.

Pacing and Timing We knew beforehand that this would be a tricky requirement to fulfil, specially considering we didn't even tackle the pacing problem. But only regarding the time the results aren't very uplifting 23% of the users considered the way how we represent time inadequate of extremely inadequate with 52% considering not negative but still insufficient. However, 88% of the users considered the graphical feedback of the precedence of the scenes extremely useful and 52% of them said they would prefer if said precedence would be done entirely by graphical means.

Scope 70% of the users considered the representation of dialogues and overall presentation of the scene content extremely adequate.

5.2 Systems tested

To better acknowledge how well our decisions proved to be the right ones, the systems implemented were tested having as base two critical aspects: *Usability* and *Adequacy*.

Scenes Regarding *Usability* the editing of scenes received a generally positive feedback from users with an average rating (in a scale of one to five) of 4,24 (standard deviation of 0,79) when evaluating the easiness of adding elements to the scene, when it comes to editing the scene the result is a little worse with an average rating of 3,82 (standard deviation of 1,12) with the main complaint (all groups except one reported this problem) being the time it took to insert new text in the plot section of a scene. Analysing *Adequacy* 70% of the users said the organization of a scene if very adequate to its purpose.

Time and Branching Regarding time, the results obtained were extremely disappointing. On one hand 88% of the enquiries said it was extremely easy to alter the time of a scene, on the other hand 70% of the users said the system used to represent time was inadequate. When it comes to branching the results were more positive with 64% of the users reporting that it was easy to create an alternative path to the story with an equivalent percentage stating that the method to create alternative paths was adequate.

Conflicts Concerning the Conflicts mechanism 88% of the users reported that this type of mechanism is important for the creation of an Interactive Story and to maintain balance. Of the users that used the Conflict system 78% said it was easy to use. However, only 55% said the system was adequate for its purpose a value we think states directly to the notion some users reported that a system like that should be more flexible and allow for a greater interaction between the participants instead of only bidding one time, a system like in Poker where you can raise was said to be preferred.

Other data Some other data emerged from the analysis of the surveys mainly on the negative side. Although for the most part the usability of the system was not a problem. The truth is 41% of the users complained about the aspect of the program stating that it should be more appealing and not so work-focused. Another problem that came up frequently was the lack of feedback regarding the amount of coins each player had, which was not explicitly stated at all time. It should also be mentioned that in terms of the method used to create the story 88% of the users considered the method to be valid and they said they would use it to create an Interactive Story if the need arose, unfortunately the results for the use of the application were slightly less positive with only 64% saying they would use StoryColla to create an Interactive Story. This validates the method we used to create the story but identified several problems relating to our application and its inefficiency to properly implement our method of authoring.

6 Conclusions

Authoring in Interactive Storytelling still proves to be a very serious task. We set ourselves to create a tool to be used in a group context capable of fulfilling the requirements already identified for authoring in IS. After the analysis of our results we can only claim a partial victory in this aspect. Although achieving 5 out of 6 requirements isn't too bad for a first approach to this type of system, the truth is that the non-compliance to all requirements makes this an incomplete tool by itself. Which is not a failure, but obviously could be better.

In terms of method we got some positive feedback and several users said they found the method to be appropriate to create Interactive Stories, however our application failed to reproduce that in a positive fashion. Concerning the previous need of programming knowledge we can safely say that our application did not require that and it did not influence in the usage of it by the specific subset of users. However, users with previous programming knowledge and some experience in IS mentioned that in certain aspects, namely the Action Scenes, the level of control should be bigger and it should not be so high-level.

Overall, we can consider that our prototype managed to achieve the objectives it set itself to in a first approach to this type of systems where work has been very scarce. However, some improvements should undoubtedly be made.

References

1. Ernest Adams. *Fundamentals of Game Design.* New Riders Publishing, Thousand Oaks, CA, USA, 2nd edition, 2009.
2. Joseph Bates. Virtual reality, art, and entertainment. In *PRESENCE: Teleoperators and Virtual Environments*, pages 133–138. MIT Press, 1992.

3. Bioware. Mass effect. Videogame, May 2008.
4. Steve Burbeck. Applications programming in smalltalk-80(tm): How to use model-view-controller (mvc), 1987.
5. Metropolitan Police Campaign. Choose a different ending. `http://www.youtube.com/watch?v=JFVkzYDNJqo`.
6. Mike Carbonaro, Maria Cutumisu, Matthew Mcnaughton, Curtis Onuczko, Thomas Roy, Jonathan Schaeffer, Duane Szafron, Stephanie Gillis, and Sabrina Kratchmer. Interactive story writing in the classroom: Using computer games. In *In Proceedings of DiGRA*, pages 323–338, 2005.
7. Chris Crawford. *Chris Crawford on Interactive Storytelling (New Riders Games)*. Number ISBN 0-13-146099-4. New Riders Games, 2004.
8. Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. *Human Computer Interaction*. Pearson/Prentice-Hall, Harlow, England, 3rd ed. edition, 2004.
9. Stefan Göbel, Rainer Malkewitz, and Felicitas Becker. Story pacing in interactive storytelling. In Zhigeng Pan, Ruth Aylett, Holger Diener, Xiaogang Jin, Stefan Göbel, and Li Li, editors, *Technologies for E-Learning and Digital Entertainment*, volume 3942 of *Lecture Notes in Computer Science*, pages 419–428. Springer Berlin / Heidelberg, 2006.
10. Juan Pablo Hourcade, Benjamin B. Bederson, Allison Druin, and Gustav Taxén. Kidpad: Collaborative storytelling for children. In *CHI '02 extended abstracts on Human factors in computing systems*, CHI EA '02, pages 500–501, New York, NY, USA, 2002. ACM.
11. Brenda Laurel. *Computers as Theatre*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.
12. Michael Mateas. The authoring challenge in interactive storytelling. In Ruth Aylett, Mei Lim, Sandy Louchart, Paolo Petta, and Mark Riedl, editors, *Interactive Storytelling*, volume 6432 of *Lecture Notes in Computer Science*, pages 1–1. Springer Berlin / Heidelberg, 2010.
13. Ralph Mazza and Mike Holmes. *Universalis*. Ramshead Publishing, 3rd edition, 2002.
14. Ben Medler and Brian Magerko. Scribe: A tool for authoring event driven interactive drama. In Stefan Göbel, Rainer Malkewitz, and Ido Iurgel, editors, *Technologies for Interactive Digital Storytelling and Entertainment*, volume 4326 of *Lecture Notes in Computer Science*, pages 139–150. Springer Berlin / Heidelberg, 2006.
15. N Montfort. *Ordering Events in Interactive Fiction Narratives*, pages 87–94. 2007.
16. Yoshiyuki Nakamura, Maiko Kobayakawa, Chisato Takami, Yuta Tsuruga, Hidekazu Kubota, Masahiro Hamasaki, Takuichi Nishimura, and Takeshi Sunaga. Zuzie: Collaborative storytelling based on multiple compositions. In *ICIDS*, pages 117–122. Springer, 2010.
17. E. Packard. *The cave of time*. Choose your own adventure. Bantam Books, 1979.
18. David Pizzi and Marc Cavazza. From debugging to authoring: Adapting productivity tools to narrative content description. In Ulrike Spierling and Nicolas Szilas, editors, *Interactive Storytelling*, volume 5334 of *Lecture Notes in Computer Science*, pages 285–296. Springer Berlin / Heidelberg, 2008.
19. Jennifer Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: Beyond Human-Computer Interaction*. Wiley, Indianapolis, IN, 2002.
20. Mark O. Riedl and R. Michael Young. From linear story generation to branching story graphs. *IEEE Computer Graphics and Applications*, 26:23–31, 2006.
21. Kimiko Ryokai and Justine Cassell. Storymat: a play space for collaborative storytelling. In *Computer Human Interaction*, 1999.
22. Yang Ting Shen and Ali Mazalek. Puzzletale: A tangible puzzle game for interactive storytelling. In *Proceedings of the 7th Conference on Advances in Computer Entertainment Technology*. ACM, 2010. To appear.
23. Nicolas Szilas, Olivier Marty, and Jean hugues Réty. Authoring highly generative interactive drama. In *In: 2nd Int. Conf. on Virtual Storytelling*, pages 37–46. Springer, 2003.
24. Loren G. Terveen. Overview of human-computer collaboration. *Knowledge-Based Systems*, 8(2-3):67 – 81, 1995. Human-computer collaboration.