Emergent Stories Facilitated

An architecture to generate stories using intelligent synthetic characters

Rui Figueiredo[1], Antonio Brisson[1], Ruth Aylett[2], and Ana Paiva[1]
rui.figueiredo@tagus.ist.utl.pt,
antonio.brisson@gaips.inesc-id.pt, ruth@macs.hw.ac.uk,
ana.paiva@inesc-id.pt

[1] INESC-ID, Avenida Prof. Cavaco Silva - Taguspark, 2780-990 Porto Salvo, Portugal
[2] Heriot-Watt University, Edinburgh EH14 4AS, UK

Abstract. We consider the issues involved in developing an architecture that is generic enough to be applied to entertainment/educational applications that convey stories through the interactions between intelligent graphical characters and a user. Issues in organizing the emergent content are discussed and two two storytelling applications that use the architecture are described.

Keywords: Autonomous agents, synthetic characters, interactive storytelling, narrative generation, interactive virtual environment

1 Introduction

Educational games are a good tool to teach people about a certain subject, reinforce development, or assist in learning a new skill. In particular, educational video games sometimes referred to as a edutainment or serious games (because they mix education and entertainment), can be seen as an effective tool for creating rich contexts, reflective of the real world that help to promote learning [1].

Traditional video games tend to limit user interactivity in order to guarantee the sequential unfolding of a predefined story. There are, however, several approaches to create game-like systems that try overcome this limitation. Some have adopted a branching narrative structure, in which a finite number of prescripted paths result from a choice made at a specific decision point [4] [7]. In other cases work has tried to cover the whole space of possible options [8], with a correspondingly combinatorial authoring problem.

All these approaches try to tackle the problem that occurs when the user has the freedom to do what he/she pleases. If we give too much control to the user the stories created through the interaction with the system might be meaningless. However, if we limit the user interactivity too much we create a system that tells one story, which is always the same and that will quickly make the user loose interest. This is especially true in an educational context where the stories are not so appealing to users as in traditional video games.

The particular type of application we are trying to achieve is one that simulates educational role-play. In educational role-play social interaction is used as the stimulus for challenging and changing existing beliefs [13] and can result in significant behavioral changes [6] making it highly relevant for social and emotional learning [2] [5].

In role-play the story is improvised rather than scripted and emerges from the interactions between the characters involved. Since the story is emergent there is no guarantee that it will convey a meaningful message. This problem is handled in traditional educational role-play through a person, usually referred to as the facilitator of the story, that is responsible for ensuring that the story develops within a certain boundary of the context originally conceived for the role-play.

Role-play is often developed as a succession of scenes, in which external events and consequences of actions within scenes may be controlled by the facilitator of the role-play. Each participant has a back-story that describes the role of his/her character, and that is communicated by the facilitator at the start of each scene. In some cases the facilitators will themselves play a character with the specific intention of shaping the emerging story in particular ways. It is through the use of these methods that it is possible to bound the scope of the emerging story around a particular educational subject.

This paper proposes an architecture for a virtual version of a human facilitator that mimics the methods used by a real facilitator. In the remainder of this paper the proposed architecture is described and two examples of its application are given.

2 Related Work

Over the pas few years the quest for creating Interactive Storytelling applications has imposed not only technical and artistic challenges, but also the challenges that derive from the cooperation between these fields. More precisely we should point out conflict between the user and the author intentions toward the story development as an example of a problem that is related to the cooperation between the artistic goal of structuring the story and the technical goal of achieving a good user experience, also called the *boundary problem*[12].

Several architectures have been proposed to tackle this challenge from which we distinguish; *Character Centered Approach*, where all the knowledge of the story structure is in the characters AI, and the *Mediation Approach*, which implements an entity, in our case a Story Facilitator(SF), that mediates the possible conflicts between user actions and the authored story.

Character Centered Approach as been used in adventure computer games and in the Interactive Storytelling prototypes by Cavazza et all.[21] based in a very strong and detailed character AI. This approach achieved good evaluation metrics such as story scalability[22], nevertheless the characters AI is confined to the author's definitions that end up guiding the user through a story limiting its interference in the outcome.

Projects that use *Meditation*, can implement several independent strategies to include the user in the story. An example is the MOE architecture proposed by Weyhrauch[16], in which the mediator distinguishes between acting upon the characters internal state, world state or the state of the interaction. In Mimesis[18][19] the *Mediation* on deciding whether to *Accommodate* user/author conflicts adjusting the author plans, or to *Intervene* causing the user conflicting action to fail. In Façade[10][9] the mediation is more subtle because the story is fragmented into different *Action Beats*, that can be interrupted and sequenced to keep up the story flow.

Using Mediation is a step toward including some user actions in the story, increasing immersion and agency. Nevertheless, in the above examples the characters autonomy is still very reduced limiting their reactions to environment changes and consequently limiting user interaction, and interference in the story outcome. Our approach described in the following section addresses this issue by implementing a *Mediation* architecture that is prepared to interact with an environment populated exclusively by autonomous characters.

3 Architecture

In educational role-play all the participants have a role which is communicated to them by the facilitator. Usually, that role is presented through a back-story (a description about the character from where the participant can interpret his/her role). After the participants figure out their role, the role-play scenes begin, and it is the role of the facilitator to decide when each scene ends, and when the next one begins. During the scenes the facilitator might intervene if he/she finds it necessary (for example, if the role-play is taking a path that is in conflict with the intended one for the role-play). The facilitator might even take the role of a character in order to guarantee that certain events happen (that might be important for the perception of the intended message to be conveyed by the role-play).

To achieve this behavior, the architecture that was conceived (Figure 1) is composed by a *Story Facilitator* agent, a *Story Memory* and a set of *Narrative Actions* that are available to the *Story Facilitator* (SF). The story is divided in episodes that roughly represent role-play scenes. Each episode is a description of where the action takes place, who is going to participate in it, what is the role of each participant, when it can be selected and when it should end, and a set of rules that describe in what situations the SF should intervene, and which action it should take during the intervention.

This architecture assumes that the environment is populated by autonomous agents and user(s) that when act produce events that can be perceived by the SF agent. Furthermore, the SF agent can change the environment through a set of actions named *Narrative Actions*.

3.1 Episodes

An episode (Figure 2) represents a part of the story that can be combined with other episodes, where each combination produces a different overall story.

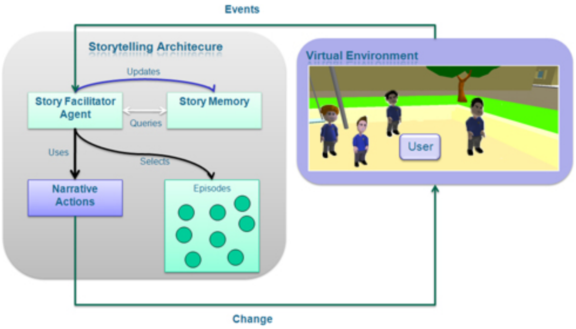


Fig. 1. The architecture

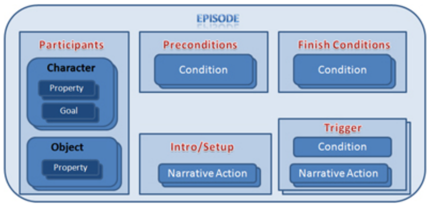


Fig. 2. Episode Structure

As the story unfolds and the SF perceives the actions that occur in the environment it populates the *Story Memory* with *Story Events* (which are descriptions of the actions the characters performed). The episodes' preconditions are tests to the *Story Events* that when satisfied indicate that the episode is a candidate for selection by the SF. The *Story Memory* also contains information that describes the characters, for example if a character is hurt or not. That information is stored as *properties* and can also be tested by the episodes' preconditions.

The finish conditions express when the SF should end the episode and select another one for the story to continue. The episodes' finish conditions are conditions identical to ones contained in the episodes' preconditions.

Each episode contains a list of the participants that take part in it. A participant can be a character or an object(prop). For each character participant the episode contains a list of its goals and properties. This can be seen as a explicit representation of their back-story, that is communicated to them as soon as the episode starts. A property for a participant is a characteristic inherent to him/her/it. For example, if it is a property about an object (prop) representing a book, the property might be called *belongs-to* and its value the name of one of the characters. In the case of character participants the episode contains a list of goals for each of them (the episode only contains the name of the character's goal, the actual implementation of the goal is in the autonomous agent that controls the character).

Each episode has an intro/setup which is a set of *Narrative Actions* the SF should perform as soon as the episode starts. The typical intro/setup for an episode is the placing of the characters in the environment.

Similarly to what happens in real educational role-play there might be situations where the SF has to intervene during an episode. Those situations are specified as *Triggers*. A trigger is composed by a condition and a set of *Narrative Actions* that the SF will perform if the trigger's condition is satisfied.

3.2 Story Memory

The *Story Memory* represents the story as how it is perceived by the SF agent. Every time a character performs an action (be it a character controlled by an autonomous agent or a user) the SF perceives the action and creates a *Story Event* (Table 1) that it stores in the *Story Memory*.

Attribute	**Description**
Subject	The ID of the agent (or user) that performed the action
Action	The name of the action
Parameters	The parameters used in the invocation of the action
Event Type	Indicates if the action is starting or finishing
Episode	The episode where the story event took place

Table 1. Attributes of a story event

Also, each time a character is placed in the virtual environment the SF records in the *Story Memory* the character's properties (Table 2) contained in the selected episode's definition.

Attribute	**Description**
Name	The name of the property
Value	The value of the property
Holder	The character or object to whom this property refers to

Table 2. Attributes of a property

Conditions Using the information stored in the *Story Memory* the SF tests if the conditions in the preconditions, finish conditions or triggers contained in the episodes are satisfied. The episode's preconditions, finish conditions and the triggers' condition are composed by the conjunctions and/or disjunctions of *Event* and *Property Conditions*, which are the two types of conditions that the SF can test.

Field	**Description**
Subject	The id of a participant(character, object or user)
Action	The name of the action
Parameters	The parameters for the action being tested
Episode Name	The name of the episode where the event was generated
Event-Type	If the event is an event that indicates that the action is starting or is finishing
Negated	If true indicates that the condition is true if there is no story event that satisfies it

Table 3. Query to a Story Event

An *Event Condition*(Table 3) tests if there is any (or none) *Story Event* in the *Story Memory* that satisfies it. All the parameters of an *Event Condition* are optional. If the fields: *subject*, *action*, *parameters* and *episode name* are not specified this indicates that they may take any value. If not specified, the *event-type* parameter indicates that we are testing for an event that indicates that the action has finished. For instance, imagine a condition that only specifies the *subject*, for example *John* and the action *Cry*. This condition will always return true after the character *John* finishes the action *Cry*.

Field	**Description**
Property Name	The name of the property
Holder	The name of the holder of the property, can be an object, an agent or the user
Value	The value of the property
Operator	One of the following operators: *Equal, NotEqual, LesserThan, GreaterThan*

Table 4. Query to a Property

The other type of condition is the *Property Condition*(Table 4). In this type of condition all the fields except the field *Operator* can be omitted, indicating that their value should not be considered. If the field, *Value*, is omitted the *Operator* field is ignored. Imagine as an example, a query to a property where we specify the *property name* and the *holder* and we omit the *value*. This query returns false only when there is no such property for that *holder*.

3.3 Narrative Actions

The *Narrative Actions* (Table 5) are the actions the SF has available to intervene while the story is unfolding. They are used during the *intro/set-up* of the episodes and when a *trigger* is selected for execution.

The *Insert Character/Object* are mainly used during the *intro/setup* of an episode to place the characters and objects on the environment. They can also

Narrative Action	**Description**
Load Scenario	This action loads a scenario(e.g. 3D model of a classroom)
Insert Character	This action inserts an agent in the current episode
Insert Object	This action inserts an object in the current episode
Narrate	Writes text on the interface. Used to simulate a narrator
Add goal	Adds a goal to a particular character
Remove goal	Removes a goal from a particular character
Act for Character	Makes a specific character perform an action
Remove Object	Removes an item from the set
Remove Character	Removes a character from the set

Table 5. Narrative actions available to the author

be used while an episode is unfolding through the use of a trigger, for example to insert a new character when a specific set of events occurs.

The *Add/Remove goal* actions are used to change the behavior of a character when certain events occur. These *Narrative Actions* can be used as "memory packets" as described in [25]. In role-play a *memory packet* is something a character is not to remember until an appropriate time in the game. Using an example taken from [25] that refers to a role-play about *Hamlet*: "...a (memory) packet meant to be opened after drinking a special kind of tea, a packet that might tell her (the character) that ... she is now passionate about botany and has forgotten all about Hamlet...". The *Add/Remove goal* action allow us to write triggers for the episodes that behave as "memory packets" as they are used in real role-play.

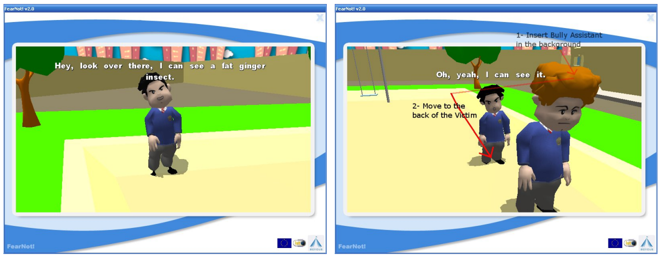
The *Act For Character* action is used to make the SF take control of a character. It is useful to ensure that one or more events take place in the story that might be important to ensure that the message to be conveyed by the role-play is understood. Furthermore, because this architecture is meant to be used with autonomous agents, which require a great deal of authoring, the *Act For Character* action is useful to create scripted behaviors (for example, to make a character walk to a particular part of the set and introduce himself) that otherwise would be difficult to create with goals and plans (if the agent architecture used for the characters is based on planning).

3.4 Triggers

As the story unfolds there might be situations where the actions of the characters and/or user(s) alone might not be enough to the progression of the story. For example, if another character is supposed to enter after a particular interaction between the characters in scene. To achieve these exogenous behaviors, when creating the episodes, we (as authors) specify the triggers that signal the moments when SF's intervention is necessary during the story. For each trigger it is necessary to specify a condition (that states when the *trigger* should be "fired") and a set of *Narrative Actions* to be executed when the trigger is selected for execution.

As an example of a *trigger*, imagine a situation where we want a character to make an appearance (*Paul*) when another (*Luke*) mocks the victim (*John*)

(Figure 3). That *trigger* has as its condition that the character *Luke* performs the action *Mock* to the character *John*, and as the *trigger's* narrative actions it has the *Insert Character* narrative actions, followed by an *Act For Character* narrative action that will make the inserted character move to a particular point of the set (behind *John*).



Hey, look over there, I can see a fat ginger insect.

Oh, yeah, I can see it.

1- Insert Bully Assistant in the background

2- Move to the back of the Victim

(a) Luke mocks John (b) Paul enters the story

Fig. 3. Trigger that is fired when Luke mocks John

3.5 Story Facilitator (Cycle)

The Story Facilitator's tasks can be described in a cycle (Figure 4) that is repeated until the story ends.

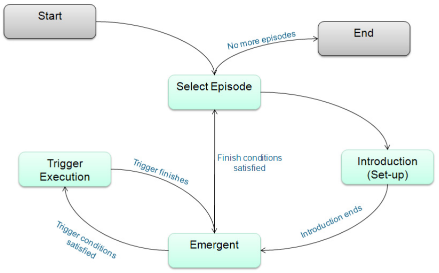


Fig. 4. The stages the episodes go through

At the very beginning (*Start*) the SF loads all the episodes for the story and puts them in the *Story Memory*. It then selects an episode that has an empty set of *preconditions* and marks that episode as *selected* so it will not be selected again.

The SF will then execute all the *Narrative Actions* contained in the selected episode's *intro/set-up*. Each time a character that is not controller by a user is inserted on the set, the SF reads that character's *properties* and stores them in the *Story Memory*, and then sends the character's goals enumerated in the selected *episode* to the agent that is responsible for controlling the character. After the *intro/set-up* of the episode ends, the SF hands the control to the agents and user(s) controlling the characters, making the story emergent from that point on.

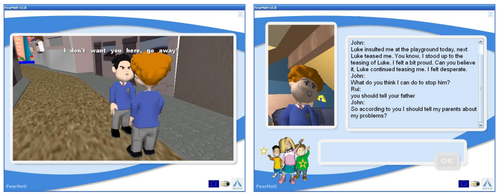
During this phase, where the agents and user(s) control the characters present in the set, the SF perceives all actions performed in the virtual environment and updates the *Story Memory* with *Story Events*. If the content of the *Story Memory* satisfies a *trigger's* condition, the SF intervenes and executes the *trigger's Narrative Actions*.

Finally, if one of the episode's *finish conditions* is satisfied by the contents of the *Story Memory* the SF ends the current episode and uses the *Story Memory* to check if there are other episodes whose *preconditions* are satisfied. If so, the SF selects one of them as the next episode. If there is no other episode that can be selected, the SF ends the story.

4 Case Studies

4.1 FearNot!

FearNot! (Figure 5) is an Interactive Virtual Environment (IVE) developed to be used as an educational tool to promote awareness about bullying behavior in schools.



I don't want you here, go away!

John:
Luke insulted me at the playground today, next Luke teased me. You know, I stood up to the teasing of Luke. I felt a bit proud. Can you believe it, Luke continued teasing me. I felt desperate.
John:
What do you think I can do to stop him?
Rui:
you should tell your father
John:
So according to you I should tell my parents about my problems?

(a) Luke bullies John (b) The learner interacts with John

Fig. 5. FearNot! application

FearNot! is inspired by role-play. Each character has a role that is related to a typical bullying scenario in a school. There are bully characters, bully assistants, victims, defenders (try to help the victim), by standers, etc. The user has the role of being a friend of the victim.

The story develops as the user is asked for advices by the victim character on how to cope with the bullying situations he/she's being victim of (there is a version with only male characters and another with only female). The victim tries to follow the advices, although he/she sometimes might not be able to.

The agents that control the characters use an agent architecture [14] [15] based on the OCC model of emotions [11]. Their motivations depend on their emotional state, which produces the perception in the user that the characters' actions fail or are succeeded, not only by the preconditions that each action has to be succeeded, but on the characters emotional state also. This is particularly important in respect to the victim character, who often "feels" fear to comply with some of the user's advices (fighting back the bully, for example).

FearNot! has roughly 40 episodes and is currently being evaluated in English and German schools.

4.2 I-Shadows

I-Shadows is an Interactive Storytelling application that takes the form of a Chinese Shadow Theater (Figure 6). In this theater the user interacts with the system by controlling a physical puppet who either is a Hero or a Villain and whose movements are interpreted by a vision system that sends that information to the autonomous characters environment.

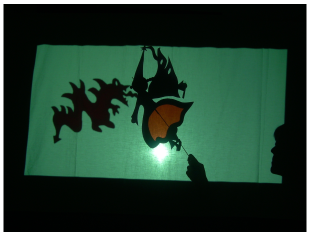


Fig. 6. Real Puppet(Fairy) interacting with a Virtual Puppet (Dragon)

This system is inspired by fairy tales, the characters are authored regarding an approximation to Propp's morphology, and the story develops as the user interacts with the other characters using expressive movements[24]. Because I-Shadows also implements exactly the same autonomous agents and Story Facilitator architectures than the above example, it allows to measure the mood development of the scene and as a consequence the moments of tension and resolution [23] of the story. Using this information and the characters relations, this system can take actions in order to potentate the development of the story flow by adding or removing characters and objects that contribute to a predefined mood development. I-Shadows is currently in user testing phase, nevertheless the experience of implementing the Story Facilitator in the system as been very encouraging and successful.

5 Conclusions and Future Work

In this paper we described an architecture inspired by educational role-play that can be applied to story-generating environments populated by intelligent synthetic characters.

We describe two applications where this architecture was used, *FearNot!* and *I-Shadows*. We believe that this architecture can easily be applied to applications that use intelligent synthetic characters in a virtual environment to create stories, as are the two systems we described.

The current version of the architecture requires a substantial amount of authoring work that can be made unnecessary if the Story Facilitator (SF) himself was gifted with planning abilities. In traditional role-playing games, such as *Dungeons&Dragons*, the story is planned by the *game-master* prior to the actual role-play by the characters. While the role-play develops the *game-master* intervenes (by changing the environment, by taking controlling of a character, etc) each time it is necessary, and as a way to coerce the role-players, to follow the story he/she had idealized. If the SF agent is capable of processing story goals and planning the story a priori, the triggers the author has to write to account for unwanted character actions that might damage the story, could be handled automatically in a mediation process similar to the one used in [17] [20]. Furthermore, the SF could automatically decide when to finish the episodes, and therefore eliminate the need for finish conditions, if the episodes were annotated with episode-level goals.

Acknowledgments

This work was partially supported by European Community (EC) under the IST RTD programme, contract IST-2001-33310-VICTEC, and is currently funded by the eCircus project IST-4-027656-STP. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein. FCT TEM DE APARECER AQUI TB? FCT TEM DE APARECER AQUI TB?FCT TEM DE APARECER AQUI TB?FCT TEM DE APARECER AQUI TB?FCT TEM DE APARECER AQUI TB?

References

1. Bednar, A.K., Cunningham, D., Duffy, T.M., and Perry, J.D. (1991). Theory into practice: How do we link? In G. Anglin (Ed.), Instructional Technology: Past, Present and Future. Englewood, CO: Libraries Unlimited, Inc.
2. Davison, J.; Arthur, J. (2003). Active Citizenship and the Development of Social Literacy: a case for experiential learning. Canterbury: Citizenship and Teacher Education.
3. Gratch J. (2000) mile: Marshalling Passions in Training and Education. In 4th International Conference on Autonomous Agents, ACM Press, June 2000
4. Gratch, J. and Marsella, S. (2001). Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. Paper presented at the Fifth International Conference on Autonomous Agents.

5. Henriksen. (2004). On the Transmutation of Educational Role-Play: a critical reframing to the Role-Play to Meet the Educational Demands. In M. Montola & J. Stenros (Eds.), Beyond Role and Play - Tools, Toys and Theory for Harnessing the Imagination (pp. 107-130). Helsinki
6. Lewin, K. (1951). Field Theory in Social Science. New York: Harper and Row.
7. Marsella, S., Johnson,W.L. and LaBore, C (2003) Interactive Pedagogical Drama for Health Interventions. In 11th International Conference on Artificial Intelligence in Education, Australia, 2003
8. Mateas, M.; Stern,A.: Architecture, authorial idioms and early observations of the interactive drama Façade. Technical report, Carnegie Mellon University, 2002.
9. Mateas M. and Stern A.: FAaDE: An experiment in building a fully-realized interactive drama. In Game Developer's Conference: Game Design Track, 2003.
10. Mateas, M.: Towards integrating plot and character for interactive drama. In Socially Intelligent Agents: The Human in the Loop, AAAI symposium, 2000.
11. Ortony,A, G. Clore, A. Collins. (1988) The Cognitive Structure of Emotions. Cambridge University Press
12. Magerko, B.: Evaluating Preemptive Story Direction in the Interactive Drama Architecture. Journal of Game Development, 2(3), 2007
13. Piaget, J. (1972). The Principles of Genetic Epistemology. London: Routledge & Keegan Paul Ltd.
14. Dias,J.: FearNot!: Creating Emotional Autonomous Synthetic Characters for Emphatic Interactions. MSc Thesis, Instituto Superior Técnico, 2005.
15. Dias, J.; Paiva, A.: Feeling and Reasoning: A Computational Model for Emotional Characters. Paper presented at the Portuguese Conference on Artificial Intelligence (EPIA), 2005.
16. Weyhrauch, P. 1997. Guiding Interactive Drama. Ph.D. diss., Dept. of Computer Science, Carnegie Mellon.
17. Saretto,C.J.; Young, M.: Mediation in mimesis Liquid Narratives. ACM Southeast Conference, 2000.
18. Young M., Riedl M., Branly M., Martin R.J., and Saretto C.J.: An architecture for integrating plan-based behavior generation with interactive game environments. Journal of Game Development, 1, 2004.
19. Michael Young R. and O. Riedl M.: Integrating plan-based behavior generation with game environments. In ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology, pages 370-370, New York, NY, USA, 2005. ACM Press.
20. Harris, J.; Young, M.: Proactive Mediation in Plan-Based Narrative Environments. Proceedings of the 6th International Working Conference on Intelligent Virtual Agents - IVA, 2005.
21. Charles, F. and Cavazza, M. and Mead, J.: Character-Based Interactive Storytelling. IEEE Intelligent Systems Vol.4 N.4 pp.17-24. IEEE Educational Activities Department.
22. Charles, F. and Cavazza, M.: Exploring the Scalability of Character-Based Storytelling. Proceedings Vol.02 pp. 872-879, AAMAS 2004. Springer.
23. Brisson, A., Paiva, A.: Are we Telling the Same Story? AAAI Fall Symposium on Narrative Intelligence Technologies, Westin Arlington Gateway, Arlington, Virginia, November 9-11, 2007
24. Brisson, A., Dias, J., Paiva, A.: From Chinese Shadows to Interactive Shadows: Building a storytelling application with autonomous shadows. Workshop on Agent-Based Systems for Human Learning and Entertainment (ABSHLE)AAMAS ACM 2007, 11-02-2007
25. Murray, J.: Hamlet on the Holodeck: The future of narrative in cyberspace (pp.117-119). MIT Press, 1998.