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. In this paper we address the issue of how to transform a set of actions and behaviors generated by autonomous characters into a meaningful and interesting story. To do that, 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. As such, issues in organizing the emergent content are discussed and a generic story facilitator (a kind of game master) is described. Finally two storytelling applications that use the architecture are described demonstrating the generality of the approach.

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

1 Introduction

Educational games are increasingly popular as a way 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 [2].

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 creating 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 [5] [8]. In other cases, work has tried to cover the whole space of possible options [9], with a correspondingly combinatorial authoring problem.

All these approaches try to tackle the problem that occurs when users have the freedom to do what they please. The dilemma is that too much user freedom may result in interactions with the system that fail to produce a meaningful story-experience. However, constraining user interactivity too tightly produces a system that tells only one story, always the same, in which the user quickly loses interest. This is especially true in an educational context where the stories are typically not so appealing to users as in traditional video games.

The type of application under investigation is a computer-based version of educational role-play. In educational role-play, social interaction is used as the stimulus for challenging and changing existing beliefs [14] and can result in significant behavioral changes [7] making it highly relevant for social and emotional learning [3] [6].

In role-play the story is improvised rather than scripted and emerges from interaction between the characters involved. Since the story is emergent there is no absolute 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 role-play. This facilitator is responsible for ensuring that the story develops within certain boundaries relating to the objectives originally conceived for the role-play.

Role-play is often developed as a succession of episodes, in which the occurrence of external events as well as the consequences of actions within an episode may be controlled by the facilitator of the role-play. Each participant has a back-story that describes the role of his/her character, communicated by the facilitator at the start of each episode. 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 topic.

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

Research in Interactive Storytelling applications not only raises technical and artistic challenges, but also specific challenges that derive from the need for cooperation between these fields. Indeed, the conflict between the user's interactional freedom and authorial intentions with respect to story development strongly relate to the artistic goal of structuring the story on the one hand and the technical goal of achieving a good user experience on the other. This has also been called the *boundary problem*[13].

The architectures that have been proposed to tackle this challenge can be divided into the *Character Centered Approach*, where all the knowledge of the story structure is in the character's AI, and the *Plot-based Approach*, in which a global action sequence forms the desired story. Both may implement an entity that mediates the possible conflicts between user actions and desired story development, though with very different intentions and outcomes.

An unmediated *Character Centered Approach* has been used in the Interactive Storytelling prototypes by Cavazza et all.[22] based in a very strong and detailed character AI. This approach achieved good evaluation metrics such as story scalability[23], 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 *Mediation*, can implement several independent strategies to include the user in the story. An example is the MOE architecture proposed by Weyhrauch[17], in which the mediator distinguishes between acting upon the characters internal state, world state or the state of the interaction. In Mimesis[19][20], acting within a plot-centred approach, *Mediation* is used to decide whether to *Accommodate* user/author conflicts by adjusting the author plans, or to *Intervene* causing a user conflicting action to fail. In Façade[11][10] the mediation is more subtle because the story is fragmented into different *Action Beats* representing all possible outcomes, 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 character autonomy is still very reduced limiting their reactions to environment changes, raising the possibility of making characters act 'out of character.' It also limits user interaction, and interference in the story outcome. Our approach the *Story Facilitator* (SF) described in the following section addresses this issue by implementing a *Mediation* architecture that is prepared to shape the outcome of an unscripted narrative by interacting with an environment populated by autonomous and affective 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 of the character from which the participant can interpret his/her role). After the participants have internalised their roles, the role-play scenes begin. It is part of the role of the facilitator to decide when each episode ends, and when the next one begins. During an episode, the facilitator might intervene if that seems necessary (for example, if the role-play is taking a path that is in conflict with that intended). The facilitator might even take the role of a character in order to guarantee that certain events happen if these are important for the intended message to be successfully conveyed by the role-play.

To achieve this behavior, the architecture that has been produced (Figure 1) contains 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 into 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) whose actions produce events that can be perceived by the SF

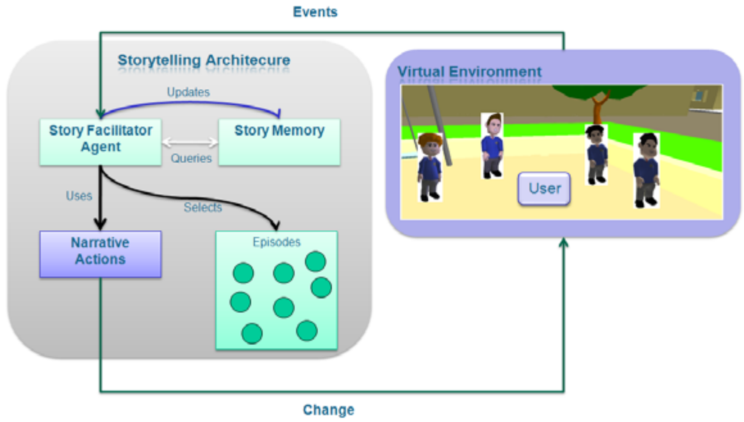


Fig. 1. The architecture

agent. The SF agent can itself 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.

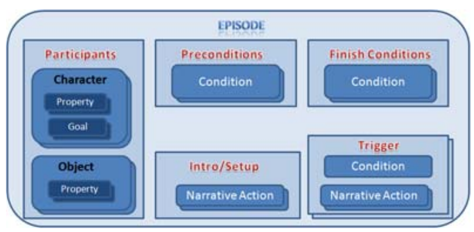


Fig. 2. Episode Structure

As the story unfolds and the SF perceives the actions that occur in the environment it populates the *Story Memory* with *Story Events* - descriptions of the actions the characters performed. The episodes' preconditions are matched against the *Story Events* and 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 whether a character is hurt or not. This information is stored as *properties* and can also be matched against the preconditions of episodes.

The finish conditions determine when the SF should end an episode and select another for the continuing story and are identical in form to the episode preconditions. Finish conditions are required because in a character-based approach there is no pre-determined sequence of actions to work through from start to finish and more abstract criteria such as the satisfaction of specific goals may be required for an episode to be considered finished.

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, communicated to them when the episode starts. A property for a participant is a characteristic inherent to him/her/it. For example, a property relating to an object (prop) representing a book might be called *belongs-to* and have as its value the name of one of the characters. For character participants the episode contains a list of goals for each of them; these are held as goal names, with the actual implementation of the goal located 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 of 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 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*.

Table 1. Attributes of a story event

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

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.

Conditions Using the information stored in the *Story Memory* the SF tests if the conditions in the preconditions, finish conditions or triggers contained

Table 2. Attributes of a property

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

in the episodes are satisfied. The episode's preconditions, finish conditions and the triggers' condition consist of conjunctions and/or disjunctions of *Event* and *Property Conditions*, which are the two types of conditions that the SF can test.

Table 3. Query to a Story Event

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

An *Event Condition*(Table 3) tests if there is any (or no) *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, consider 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*.

Table 4. Query to a Property

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*

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.

Table 5. Narrative actions available to the author

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

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

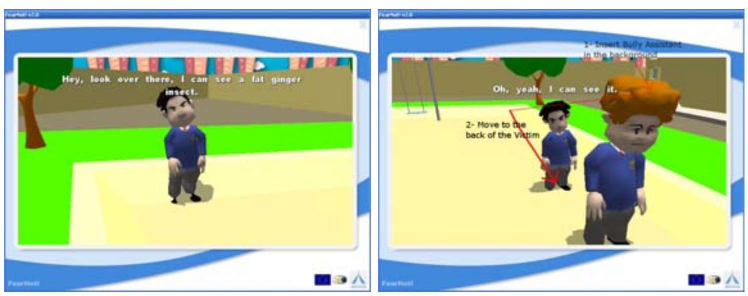
The *Add/Remove goal* actions are used to change the behavior of a character when certain events occur. These *Narrative Actions* can be used as "memory packets" as described in [26]. In role-play a *memory packet* is something a character is not to remember until an appropriate time in the game. Using an example taken from [26] 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 allow the SF to take control of a character. It could be used to ensure that one or more events take place in the story so 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 may be used 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 in an agent architecture based on planning.

3.4 Triggers

As the story unfolds there may be situations where the actions of the characters and/or user(s) alone are not enough to progress the story. An example would be where another character is supposed to enter after a particular interaction between the characters already in a scene. To achieve such exogenous behaviors, we (as authors) specify the triggers that signal the moments when SF's intervention is necessary during the story when creating the episodes. 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*).



(a) Luke mocks John (b) Paul enters

Fig. 3. The trigger 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.

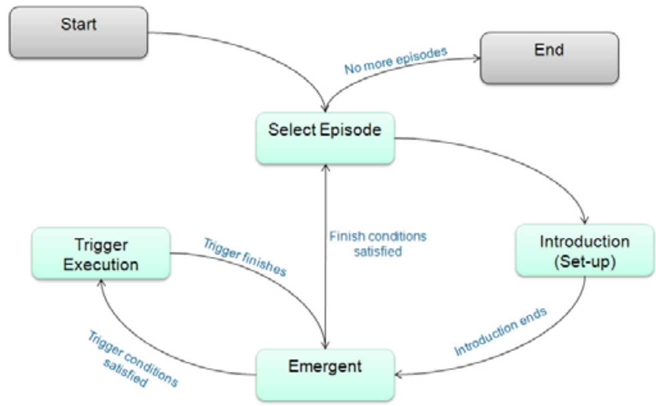


Fig. 4. The stages the episodes go through

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 controlled 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 listed 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 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 3) is an Interactive Virtual Environment (IVE) developed to be used as an educational tool to promote awareness about bullying behavior in schools.

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 (who try to help the victim), bystanders, etc. The user has the role of 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 or she is experiencing. One version has only male characters and the other only females. The victim tries to follow the advice, although sometimes they might not be able to.

The agents that control the characters utilise the FAtiMA agent architecture [15] [16] based on the OCC model of emotions [12]. Their goals depend on their emotional state, making the user aware that the actions characters carry out and whether those actions succeed or not is related to how they feel, not just to logic. This is particularly important in the case of the victim character, who often "feels" too much fear to comply with some of the user's suggestions, for example to fight back against the bully.

FearNot! has roughly 40 episodes that can be selected by the SF and has just been 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. In this theater the user interacts with the system by controlling a physical puppet that is either a Hero or a Villain and whose movements are interpreted by a vision system that sends that information to the autonomous character controllers. This system is inspired by fairy tales and the characters are authored approximately according to the morphology of Vladimir Propp. The story develops as the user interacts with the other characters using expressive movement[25]. Because I-Shadows also implements the same autonomous agents and Story Facilitator as in the FearNot! case, it allows the system to measure the emotional development of the scene and as a consequence the moments of tension and resolution [24] of the story. Using this information and the relations between characters, the system can carry out actions in order to shape the story flow by adding or removing characters and objects that contribute to a predefined mood development. I-Shadows is currently in its user testing phase, however 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 could be reduced if planning abilities were added to the Story Facilitator (SF). In traditional role-playing games, such as *Dungeons&Dragons*, an outline story is planned by the *game-master* prior to the actual role-play by the characters[1]. As the role-play develops the *game-master* may intervene in order to shape the story according to the initial conception, for example by changing the environment, by determining the outcome of user actions, by taking control of a character, etc. If the SF agent is capable of processing story goals and planning story unfolding, 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 [18] [21]. Furthermore, if the episodes were annotated with episode-level goals, the SF could automatically decide when to finish the episodes, eliminating the need for specific finish conditions.

Acknowledgements

This work was partially supported by the 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.

References

1. Aylett, R.S; Louchart, S; Tychsen, A; Hitchens, M; Figueiredo, R. and Delgado Mata, C. (2008) Managing Emergent Character-Based Narrative. Proceedings, INTETAIN 2008, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Jan 2008
2. 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.
3. Davison, J.; Arthur, J. (2003). Active Citizenship and the Development of Social Literacy: a case for experiential learning. Canterbury: Citizenship and Teacher Education.
4. Gratch J. (2000) mile: Marshalling Passions in Training and Education. In 4th International Conference on Autonomous Agents, ACM Press, June 2000
5. 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.
6. 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).
7. Lewin, K. (1951). Field Theory in Social Science. New York: Harper and Row.
8. 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

9. Mateas, M.; Stern,A.: Architecture, authorial idioms and early observations of the interactive drama Façade. Technical report, Carnegie Mellon University, 2002.
10. Mateas M. and Stern A.: FAaDE: An experiment in building a fully-realized interactive drama. In Game Developer's Conference: Game Design Track, 2003.
11. Mateas, M.: Towards integrating plot and character for interactive drama. In Socially Intelligent Agents: The Human in the Loop, AAAI symposium, 2000.
12. Ortony,A, G. Clore, A. Collins. (1988) The Cognitive Structure of Emotions. Cambridge University Press
13. Magerko, B.: Evaluating Preemptive Story Direction in the Interactive Drama Architecture. Journal of Game Development, 2(3), 2007
14. Piaget, J. (1972). The Principles of Genetic Epistemology. London: Routledge & Keegan Paul Ltd.
15. Dias,J.: FearNot!: Creating Emotional Autonomous Synthetic Characters for Emphatic Interactions. MSc Thesis, Instituto Superior Técnico, 2005.
16. Dias, J.; Paiva, A.: Feeling and Reasoning: A Computational Model for Emotional Characters. Paper presented at the Portuguese Conference on Artificial Intelligence (EPIA), 2005.
17. Weyhrauch, P. 1997. Guiding Interactive Drama. Ph.D. diss., Dept. of Computer Science, Carnegie Mellon.
18. Saretto,C.J.; Young, M.: Mediation in mimesis Liquid Narratives. ACM Southeast Conference, 2000.
19. 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.
20. 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.
21. Harris, J.; Young, M.: Proactive Mediation in Plan-Based Narrative Environments. Proceedings of the 6th International Working Conference on Intelligent Virtual Agents - IVA, 2005.
22. 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.
23. Charles, F. and Cavazza, M.: Exploring the Scalability of Character-Based Storytelling. Proceedings Vol.02 pp. 872-879, AAMAS 2004. Springer.
24. 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
25. 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
26. Murray, J.: Hamlet on the Holodeck: The future of narrative in cyberspace (pp.117-119). MIT Press, 1998.
27. Aylett, R. and Louchart, S. and Dias, J. and Paiva, A. and Vala, M. and Woods, S. and Hall, L.:Unscripted Narrative for Affectively Driven Characters IEEE Computer Graphics and Applications, Vl. 26(3), 2006. issn: 0272-1716, pp. 41-52. IEEE Computer Society,Los Alamitos, CA, USA