Shaping Emergent Narratives for a Pedagogical Application

Rui Figueiredo(1), João Dias(1), Ruth Aylett(2),
Sandy Louchart(2), Ana Paiva(1)
INESC-ID and IST(1),
Rua Prof. Cavaco Silva, Porto Salvo, Portugal

Heriot-Watt University(2),
Edinburgh EH14 4AS, UK
{rui.figueiredo, joao.assis}@tagus.ist.utl.pt,
ruth@macs.hw.ac.uk, sandy@macs.hw.ac.uk, ana.paiva@inesc-id.pt

Abstract

In this paper we explore an alternative approach to narrative generation in a pedagogical context where the story emerges from the interactions between the characters of a story in a 3D environment. Relying on previous work where the autonomy of characters was explored to generate emergent narratives, in this paper we present the emergent episode, a structure that can be used to organize the story. Also, a special agent, named Story Facilitator is presented. This agent has the task of sequencing the episodes in such a way that produces a coherent emergent story.

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

1 Introduction

Stories are not only part of us, of our memories and our lives, they are also what gives us interpretations of our world, make us learn, and provide us the context for the events we constantly observe in our environment. That's

why many learning situations are not only attached to stories, but promoted by them. In particular, when learning brings about changes in attitudes, or development of social skills, stories are seen as a good and alternative way to lead such changes in behavior.

This natural aspect of stories, is now considered not only in the current classroom activities, but has the potential to be part of educational software. Moreover, the use of virtual environments that can generate narrative can be beneficial to entertain, train or educate their users. Software environments are now used for learning with interesting and interactive story-like experiences. Believable agents, that behave like actors on a stage became now part of these systems.

However, to develop such systems it is necessary to solve several problems, and in particular the fact that authors need to be able to influence a story that is interactive and sufficiently flexible for the users to interact with. However, in most cases the author has to specify all of the actions that the characters perform, as well as predict a great number of different situations that might occur due to user intervention.

In this paper we present an alternative approach to the most common ones found in interactive narrative environments [13, 3, 4, 8, 9] and we are using it to develop a narrative learning environment. In our approach the virtual environment is inhabited by autonomous agents that control the characters of the story. In the line of the work proposed in [12], characters are partially influenced in the actions they carry, by the way they were parameterized. It is up to the author to create the personality of these characters by defining the agents' goals and personality traits which will cause them to perform in a way that is in accordance with the author's intentions when they interact between themselves and with the user.

Bellow we describe the narrative learning environment we've been working on, FearNot!, and then we briefly describe the agents and the narrative generation system.

2 FearNot!

FearNot! is an application created for an European-funded project named VICTEC[11] which tells a story with an anti-bullying message to children. The story takes place in a virtual environment, shown in figure 1, inhabited

Figure 1: FearNot!

by synthetic characters that interact with the children by asking them for advices.

In terms of story generation, FearNot! differs from the more used approaches [5, 9, 13, 2] because of its agent architecture [7]. In FearNot! the story emerges from the interactions between the agents and the user, and between themselves, thus creating an emergent narrative [1]. Within this approach, the author does not relinquish responsibility for the overall direction and shape of the story. However, rather than scripting agent actions, authoring becomes both the definition of the story world, character personalities and goals, and the definition of constraints which shape the emergent narrative, in this case using a story facilitator discussed below.

The agent architecture developed for FearNot! includes emotion in the reasoning process [6, 7] and is based on a continuous planner where emotions play a central role.

The emotion notion used is based on the OCC cognitive theory of emotions [10], where emotions are defined as valanced (good or bad) reactions to events. The assessment of this relationship between events and the character's emotions is called *appraisal* process which is made regarding the agent's goals, standards and attitudes. Goals represent world states that the agent desires to attain. Standards refer to ethics and social and moral standards and attitudes represent the agent's preferences and dispositions towards objects or people.

The agents in FearNot! have two types of goals, *active-pursuit* goals and

interest goals. Active-pursuit goals are the goals that the characters actively try to achieve, such as going to a dentist appointment. Interest goals represent goals that a character has but does not pursuit, for example avoiding to get hurt. Each goal has a set of emotions associated with it (the prospect based emotions of the OCC theory), the two most important emotions are *hope*(of achieving the goal) and *fear*(of not achieving it), these emotions represent the importance of the goal to the agent in the sense that the goals generating the strongest emotions are the ones that require more attention from the agent.

In addition to the actions that are executed by the agent in order to fulfill a certain goal, agents in FearNot! have reactive behaviors that are generated through their *action tendencies*. An *action tendency* is an action that is triggered when the agent is in a certain emotional state and when a particular event occurs.

By defining the agents' goals and *action tendencies* the author builds the characters' personalities. Just like in cinema where a character is identified by its quirks and objectives, in FearNot! these are modeled as *action tendencies*, goals and emotional parameters. These features are then used in real time as a way for the story to emerge.

3 Narrative Generation

In FearNot! stories emerge from the actions of the agents(characters) that form the cast, meaning that an emerging story is consistent with their roles and personalities. However, it is important to note, especially in a pedagogical domain, that the author must be able to constrain this process such that the emergent narrative meets their high-level authoring objectives. These authorial objectives are not espressed as detailed sequences of low-level actions as in a scripted system, but must lie at a more abstract level: for example that certain types of events and actions should happen somewhere in the story, or that particular encounters between specific characters should take place in a certain setting. The author has the ability to define the characters' personality by defining their initial objectives and personality traits, but also needs the ability to specify actions exogenuous to the actions of the characters: entrances of particular characters, or, in the episodic structure being considered here, the location, cast and character objectives of successive episodes.

Name	A unique name that identifies the episode
Set	The set is the location on the virtual environment where the events of this episode will take place.
Characters	The characters of the story are defined through a set of properties like their name, position on the set, and all the properties the author may want to include to describe the character.
Preconditions	The preconditions are a set of conditions that specify when is the episode eligible for selection.
Goals	Character goals that are communicated to the agents in this particular episode.
Triggers	A trigger is a condition that when satisfied will cause the execution of a set of *narrative actions* (the concept of narrative action will be described later).
Finish Conditions	The finish conditions are a set of conditions similar to the preconditions that when satisfied indicate that the episode is finished.
Introduction	This section of the definition of the episode is composed by a set of *narrative actions*.

Table 1: The several elements that the author defines in each episode

In order to animate the back stories that are an essential requirement for richly motivated characters, the author may require a fixed introductory scene, for example, making a certain character walk to a specific position in the set and then talk to the camera. In these cases the emergent behavior of the characters must be switched off. The author has specific sections of the episodes, described in the next section, where he can create these specific behaviors.

3.1 Story Organization

In the proposed Framework a story is organized in episodes. An episode represents a part of the story that can be combined with other episodes, with each combination creating a different overall story. Each episode contains information that allows contextualization of the part of the story it represents, together with information that allows the system to know at what point each episode should end. An episode is defined by the elements show in Table 1.

It is the sequencing of these episodes that makes up the story, as each episode

Insert Character	This action allows the author to insert a character in the current episode.
Insert Object	Similar to the *Insert Character Action* but applied to objects.
Change Camera	Changes the perspective of the camera.
Narrate	Allows the author to write text to the interface
Change Story Mode	Changes the interface.
Act for Character	Makes a specific character execute an action specified by the author.
Play audio	Plays a music file.
Remove Object	Removes an object from the set.
Remove Character	Removes a character from the set.

Table 2: Narrative actions available to the author

will specify the agents' goals, which will cause the associated characters to play their part in the episode. It is important to note that the goals of the characters are not defined in each episode. The goals are already defined in the character personality. In each episode a subset of the goals is selected and sent to the agents.

The actual sequencing of the episodes is responsibility of a special agent called *Story Facilitator*.

3.2 Narrative Actions

A narrative action is an action available to the author that he can use while defining the episodes. These actions can be used in the *Introduction* (set up the scene) of the episode, where they can be used to insert the characters of the story, set the camera to a particular position or narrate some text in the interface. They can also be used in the *Triggers*, as described bellow.

A trigger is a rule that is fired when a condition is verified, the firing of a trigger causes the execution of a set of narrative actions. The conditions can include tests to actions the user may have performed. The narrative actions available to the author are listed in table 2.

These actions allow the author to specify an introduction to each new episode (although it is possible that an episode does not have an introduction), and together with the trigger mechanism the author can specify specific situation where the system should intervene, for example it is possible to write a *trigger*

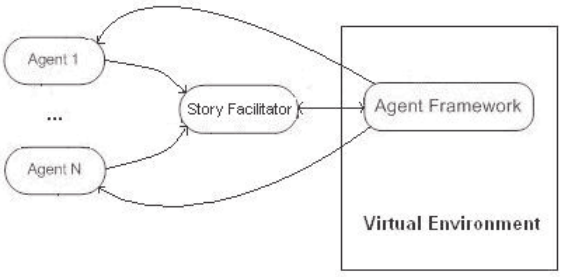

Figure 2: Communication between the Story Facilitator, the framework and the other agents

rule that adds a character to a particular scene when a character (or the user) performs a certain set of actions. Furthermore, the author can also specify text to be written on the interface if a certain situation arises in the virtual environment. As such, together with the trigger mechanism it is possible for the author to specify in what situations intervention from the system is necessary and what should be done in those situations.

3.3 Story Facilitator

Story Facilitator is the name of the agent that has the function of controlling the unfolding of the story. It has special privileges that allow it to know everything that happens in the virtual environment, having also the ability to perform special actions that affect the environment.

In figure 2 we can see that the Story Facilitator monitors all the messages between the agents and the agent framework, as well as all the messages generated within the framework back to the agents. This allows the Story Facilitator to know which actions the agents intend to execute.

The Story Facilitator keeps tracks of all the *events* that occur in the virtual environment. An *event* in this context is an action from one of the characters (or the user) in the situation it occurred. This memory of the *events* is used so that the author can specify the conditions used in the preconditions and finish conditions of the episodes, as well as on the conditions for the *triggers*.

The system behavior when supervised by the Story Facilitator can be represented as a state machine, as show in figure 3.
In the *episode selection state* one episode is selected from the group of

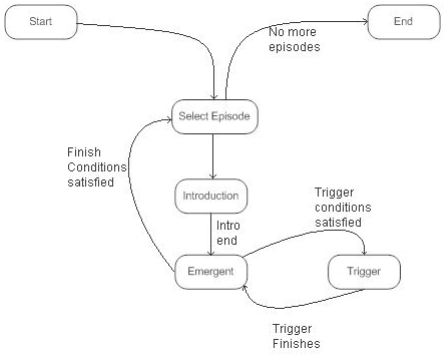

Figure 3: Different states when running an episode

episodes that have their preconditions satisfied. The criteria for deciding which episode to select is still an open issue, although we are considering the adoption of an approach similar to the one used in *Façade* [9]. Another possibility would be to use the emotional state of the characters in the story to select the most appropriate episode to play next.

After the selection of the episode, it's *introduction* (set up) is played. Such *introduction* is composed by *narrative actions* and typically includes actions such as narrating an introductory text or inserting the characters on the set.

After the *introduction* the next step is the *emergent state* where the goals of the characters dictate the unfolding of the story. It is in this state that tests, to check if there is any trigger eligible to be fired, are made. If indeed there are triggers that can be fired, the trigger that has the higher priority (the priority is defined by the author) is selected for execution.

The *trigger state* is similar to the *introduction state*, in the sense that when the system is in this state, a set of *narrative actions* is executed in order, and when they finish, the system goes back to the *emergent state*.

When the episode ends (when its "ending conditions" are satisfied) the system goes back to the *episode selection state* and is there are no more episodes, the story finishes.

As an example we will describe one very simple episode taken from a story used in FearNot!. For the sake of simplicity the goals of the characters are only mentioned, the process of authoring the goals is out of the scope of this

paper(it can be found in [6]).

The story concerns a character named John (a victim) that is bullied by Luke (the bully). This episode is used as one of the possible endings for the story, where John (the victim) asks for help to a friend named Paul (the helper) that agrees to help him. After this request, the bully will appear in the set and try to mock the victim, but Paul doesn't allow it.

At authoring time, the author has to include three characters in an episode: a bully (Luke), a helper (Paul) and the victim (John). The *preconditions* for the episode are: the user (the child using the system) must have advised the victim to tell a friend about his bullying problem. The goals of the characters are, for the victim: making friends and asking for help; for the helper: helping the victim; and for the bully: bullying anyone, in particular the victim.

At run time, the system starts by playing the *introduction* of the episode where John and Paul are placed on the set. In the *emergent* part of the episode, their goals will cause them to interact and eventually John will accept the invitation from Paul to help him, this event (arising from the autonomous minds of the characters) is detected by a trigger which will lead the Story Facilitator to insert Luke on the set. Luke will then mock John which will cause Paul to protect him, making the Luke go away. When this happens the episode comes to an end, given that the *ending condition* for the episode is detected.

4 Conclusions and future work

This paper describes the on-going efforts to develop an alternative approach to narrative generation, where the characters of the story are controlled by autonomous agents with reasoning capabilities. In this approach we intend to provide a structure, the emergent episode, that allows the author to benefit from all the advantages of an emergent approach while still being able to build a coherent story.

The next step in this work consists essentially in the development of a greater number of generic episodes and with more characters interacting simultaneously. This will allow for the generation of many different stories, thus allowing us to evaluate the quality of this approach in terms of authoring effort and coherence of the narrations generated.

References

[1] Ruth Aylett. Narrative in virtual environments - towards emergent narrative. *AAAI Fall Symposium, Thechnical Report FS-99-01*, 1999.

[2] Kevin M. Brooks. Do story agents use rocking chairs. *ICM*, 1997.

[3] Mark Cavazza, Ruth Aylett, Kerstin Dautenhahn, Clive Fencott, and Fred Charles. Interactive storytelling in virtual environments: Building the holodeck. *VSMM*, 2000.

[4] Mark Cavazza, Fred Charles, and Steven Mead. Agent's interaction in virtual storytelling. *IVA*, 2001.

[5] Fred Charles, Steven Mead, and Marc Cavazza. Character-driven story generation in interactive storytelling. *VSMM*, 2001.

[6] João Dias. Fearnot!: Creating emotional autonomous synthetic characters for emphatic interactions. Master's thesis, Instituto Superior Técnico, 2005.

[7] João Dias and Ana Paiva. Feeling and reasoning: A computational model for emotional characters. *EPIA*, 2005.

[8] Michael Mateas and Andrew Stern. Architecture, authorial idioms and early observations of the interactive drama façade. Technical report, Carnegie Mellon University, 2002.

[9] Michael Mateas and Andrew Stern. Façade: An experiment in building a fully-realized interactive drama. *Game Developers Conference*, 2003.

[10] A. Ortony, G. Clore, and A. Collins. *The cognitive structure of emotions*. Cambrige University Press, UK, 1988.

[11] Ana Paiva, Daniel Sobral, João Dias, Jonathan Pickering, Marco Vala, Ruth Aylett, and Sandy Louchart. 6.4.1: First prototype of the bullying demonstrator. Technical report, VICTEC, 2003.

[12] D. Sobral, I. Machado, and A. Paiva. Managing Authorship in Plot Conduction. *Proceedings of the 2nd International Conference on Virtual Storytelling - ICVS*, 2003.

[13] Michael Young, Mark Riedl, Mark Brandly, Arnav Jhala, R. Martin, and C. Sareto. An architecture for integrating plan-based behavior generation with interactive game environments. *Journal of Game Development*, 2004.