Driving stories, benefits of properties analysis

A. Prigent, R. Champagnat and P. Estraillier

L3i, Université de La Rochelle
Pôle Science et Technologie
17 042 La Rochelle Cedex 1
France

Tel : +33 546 458 284
Fax : +33 546 458 289
{armelle.prigent,ronan.champagnat,pascal.estraillier}@univ-lr.fr

September 30, 2005

Abstract

Emergent narrative means that it is the player that unfolds the narrative and then creates its own story. But game should control the drama in order to modify the player emotion through the game by enabling or disabling actions. In the framework of an adaptative system that manages player actions, this paper defines a formal model for interactive narration based on Linear Logic. We introduce a new kind of properties of a narration. The verification of those properties allows ensuring the quality of a game scenario. They concern the design of games as well as game execution. For each property prospects for validation are discussed.

Keywords: properties analysis, validation, game modelling, interactive narration

1 Introduction

Narration on video games is specific and design of narration based on movie's like approaches are not satisfactory [Juu99]. As the narration part increases considerably since the beginning of games development, it leads the designers to adopt software development methodologies in order to improve the quality of software product. The narration is then considered as a complex system that should be improved.

Of course, verification methods differ depending on the kind of narration. Jenkins proposes a classification of the narration depending on its importance in the gameplay [Jen03].

J. Juul in [Juu04] shows that time running is very different from movie. Actually, in games, the story is built by the player. For instance a player can save and restore a story many times. But, he also has the ability to repeat a story in order to modify the end or to avoid some mistakes he has done before. It is commonly said that experiences created conflict that the players had to work to resolve in their own favour. This conflict challenges the players, creating tensions as the work to resolve problems and varying levels of achievement or frustration.

Here we only interest ourselves to games based on emergent narratives (*i.e.* the game space enable the player to produce its own story). Our main goal is to derive an adaptative system that controls the unfolding of a story, in emergent narrative, by bounding the player ability. Approaches based on graph analysis are faced to state space explosion problem. As a consequence our work focuses on properties analysis and verification method for "on-the-fly" analysis. This paper gives prospects in interactive narration analysis in order **to improve the quality of narration in narrative games**.

Drawbacks. *Emergent narrative* means that it is the player that unfolds the scenario and, then, creates its own story. However, if the player is free to develop the story, he may lead to unexpected end or go right through the end. The whole set of possible stories in the game has to be validated to avoid this kind of situation. As scenarios are getting more and more complicated their design become close to complex system's design. As a consequence **the introduction of formal methods for game scenarios development is a way to improve the game quality**.

[VN03] underlines the absence of techniques of analysis within the framework of games. Several teams face to interactive narration. We can quote the *"Liquid narration Group"* [You03a, YRB$^+$03, You03b, RY03], the *"Zero game studio"* [Ela02, Lin02], the university of Salford [AL04, LA04c, LA04a, LA04b], the university of Michigan [ML03], the university of Teesside [CLM$^+$03], the Oz project [Mat02]. All these works concern the difficulty in generating a coherent narration of game (compared to the rules of the game and the scripting) according to the player interactions.

Contribution. In the case of emergent narrative, the player is free to run its own story according to a set of rules and narrative elements proposed by the game. Our purpose is to propose a method that helps to **ensure the quality of narration** in this kind of game.

In this paper, we argue that the set of properties that must be verified during the game can help for running an interesting game for player, whatever his experience in gaming or his knowledge of the concerned game is.

Indeed, the quality of narration is based on a set of properties. Those properties will concerned its ability to **run a correct game** (for example, is it ending correctly?), its **drama intensity** (is there enough tension and fun during the game). A first validation step is done during the game design step, where some classical properties will be verified.

Those properties will be verified at each important step in the game execution and the adaptative system will take some decisions to modify the game unfolding in order to make those properties verified.

Outline. This paper is divided into three parts. The following section gives definition for modelling game an finishes by introducing a formal model for interactive narrative game based on Linear Logic.

The second section presents our architecture for driving a game narrative unfolding. It is based on five agents managing, checking and modifying the narration model.

Finally the last section presents properties of games, with regard on the narration. These properties concern the design of game as well as the control of game (execution properties). The design of game answers the question of the benefits of classical properties on game narration. Execution properties are specific and new properties concept are introduced. Then prospects for validation of properties are presented. It begins by giving the framework of our validation requirements and follows by presenting approaches to perform validation.

2 Modelling of interactive narrative games

Let us begin by modelling interactive narrative games. Game model is not commonly defined. As a consequence we will begin by defining a model for game keeping in mind that our objective is to derive a model for narrative. Then this model for narrative is given.

2.1 Formal description of interactive drama

Related works. The *Artificial Intelligence Laboratory*, from Michigan university, works on the notion of interactive drama (see [ML03]). It is defined as an attempt to use a computer to tell a story where the user is the main character. The deriving story should be affect by the user interactions. It consists in narrative games within the player interact and then modify the story. This architecture is based on the definition of a story director agent in charge of guaranteeing the consistency of the story by: checking events by driving the story in order to avoid to compromise the future (respecting temporal constraints); using a predictive model to have detect inconsistent stories before they arise; modifying the behaviour of characters in a way to force the user to return to one of the story (included in the narrative).

Another project is on interactive drama tension (ID tension project). It is based on managing the drama tension by using a narration agent. The aim is to generate interacting story [Szi99]. The interactive drama tension is made up: Story world (it contains the state of the world as it is currently described by the story); Narrative logic (it contains rules on how the world of the story can evolve); Narrator (it decides which action(s) should be proposed to the user); User model (it is used by the narrator to choose the action according to its belief on user); And finally theatre (it manages the interaction with the user)

Game model Let us propose a definition of game. [aCSaSH04] gives the following structure for games:

- **Objectives**: they define what players are trying to accomplish within the rules of the game.
- **Procedures**: procedures are the methods of play and the actions players can take to achieve the game objectives.
- **Rules**: define game objects and define allowable actions by the players.
- **Resources**: assets that can be used to accomplish certain goals.
- **Conflict**: emerges from the players trying to accomplish the goals of the game within its rules and boundaries (obstacles, opponents, dilemmas).
- **Boundary**: is what separate the game form everything that is not the game.
- **Outcome**: expected end of game.

We are focusing on scenario analysis, thus the model presents elements of game only in a narrative point of view (see figure 1). Resources and characters of game are only described by their behavioural aspects and not their graphical ones.

A game is made up a map, a rule set and a scenario. The map contains boundaries which represents the geographical reachable localisation for characters. The game takes place inside those boundaries. Outside boundaries we can found visualisation elements that cannot be reached or modified by the game execution.

Resources and characters are located on the map. A *resource* can be *active* (with an associated behaviour) or *passive* (its state remains constant throughout game). Independently it is *active* or *passive*, it can be *local* (it belongs to only one character) or *shared*

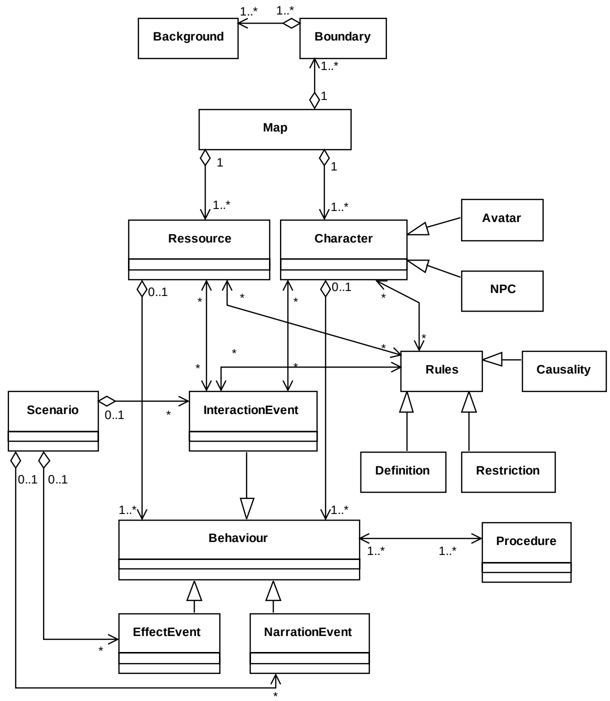


Figure 1: UML class diagram

(this resource can be used by whole or part of the character and is support of an interaction). Resources and characters are made up behaviours. Behaviour is described by a set of events. Events are of three kinds:

- **Interaction event** is an event that implies more than one element (resource or character).
- **Effect event** is triggered by condition of the game unfolding and element's state.
- **Narration event** is directed by the scripting. They are part of the game scenario. The game designer describes some events that could appear in the game.

A procedure describes how the player launches behaviour by means of game controls (mouse click, etc.).

A scenario is made up of interaction, effect and narration events. It should be pointed out that those events are not ordered. The game unfolding gives an order of event (or subset of events, they are not all implied by the game) which represents a story. This story is driven by player actions.

Rules define allowable actions and logic of events during the execution of the story. Involved objects and characters are also mentioned. Boundaries of the game have to be fixed. Three kinds of rules can be distinguished:

- **Definition**, which defines resources and characters.
- **Restriction**, which restricts some behaviour.
- And **causality** that defines effects of the player actions.

A model for game, with respect to the narration point of view, has been introduced. Let us see, in the sequel, how narration is formally modelled.

2.2 Modelling of scenario

This section deals with a modelling technique of scenario. It is based on Linear Logic (commonly said Logic for resources) and is presented in detailed in [CPE05]. We will give a quick overview in the sequel.

A scenario is modelled by the following 6 items:

- A **Resource** is an asset that can be used to accomplish certain goals. It is modelled by an atom of Linear Logic (R).
- A **Character** is an entity involved in events of the scenario insofar as it is characterised by its own events and prone to interact. It is modelled by an atom of Linear Logic (C).
- **Event** is a non preemptive atomic element of a history. Its formal representation is a linear implicative formula ($A \otimes C \multimap B \otimes C$). Two kind of events can be distinguished: *Player events*, generated by an input of the game (*e.g.* crossing a door); and *director events* which arise within the framework of the control of the game unfolding (*e.g.* appearance of a dragon when the player enters a room).
- An **outcome** is an expected end of the game (O).
- In existing works there is a confusion between terms of narrative and history. In the sequel a narrative is something which represents the unfolding of a game. A **narrative** is an ordered sequence of events. It is modelled by a sequent ($C, R, E, !F \vdash O^{\oplus}$).
- A **Scenario** is the set of all ordered sequences of events. This corresponds in writing a proof of a sequent.

Proving a sequent consists in rewriting a sequent, by introducing the definition of connectors, until it is identity. There is not a single proof and the order connectors are simplified gives the order of event in the story. Then writing all the possible proof of a sequent corresponds to write all the feasible stories of a narration (see [CPE05] for further details).

We would like to emphasis that a sequent gives the narrative framework and it is defined by the character ability. And that a proof represents a story (driven either by the player and/or the game).

3 An architecture for adaptative execution

We described here a method for driving the game unfolding based on the validation of a set of properties during the game execution. This approach is based on a specific architecture that allows analysing the game unfolding and performing modifications in this execution. We have to observe player activity, to analyse its behaviour, to verify the ongoing executions and to modify some elements in the scenario. During the execution, this architecture uses the formal model of the scenario. Moreover, it is able to modify this formal model for adapting the unfolding to the player behaviour.

This adaptative system is based on five agents: observation agent, rule guardian agent, analysis agent, scripting agent and director agent which are defined as:

- **Observation agent.** This module is in charge to capture player behaviours. For example, the player chooses to open a door by clicking on it or by pressing space bar key on its keyboard. Observation agent interprets this action on controls (i.e. the procedures of the game) and translates it in a player event corresponding to this player choice.

- **Rules guardian agent.** The game is delimited by a set of rules and they have to be respected. Rules guardian agent receives the choice forwarded by observation agent and checks it to guarantee that it respects the set of game rules.
- **Analysis agent** This agent is the one that makes "on-the-fly" verification during game execution. It receives player event from rules guardian agent and verifies that player event will not produce some errors on the ongoing execution of game. Its aim is to perform some local verifications of scenario during a particular execution. Those verifications concern for example the availability of resources, the possibility for player to reach its objective, the running of conflicts and verify that the correctness of outcomes. It produces a set of erroneous narratives that do not respect those points.
- **Scripting agent.** This agent is the one in charge to analyse the set of erroneous narratives produced by the analysing agent and proposes narratives adaptation in order to make this set of erroneous narratives unreachable. Scripting agent defines a new set of possible events. This agent uses the formal model of scenario described in linear logic, the set of narratives entities as rules, conflicts, resources.
- **Director agent.** This agent chooses a set of relevant events to execute among the set of available events produced from the scripting agent. It can choose first to allow player to perform an available action, or it can choose to perform a narration event in order to unfold the narrative. This director agent is the one in charge to modify the formal model of scenario and to run the execution forward.

The figure 2 represents this architecture.

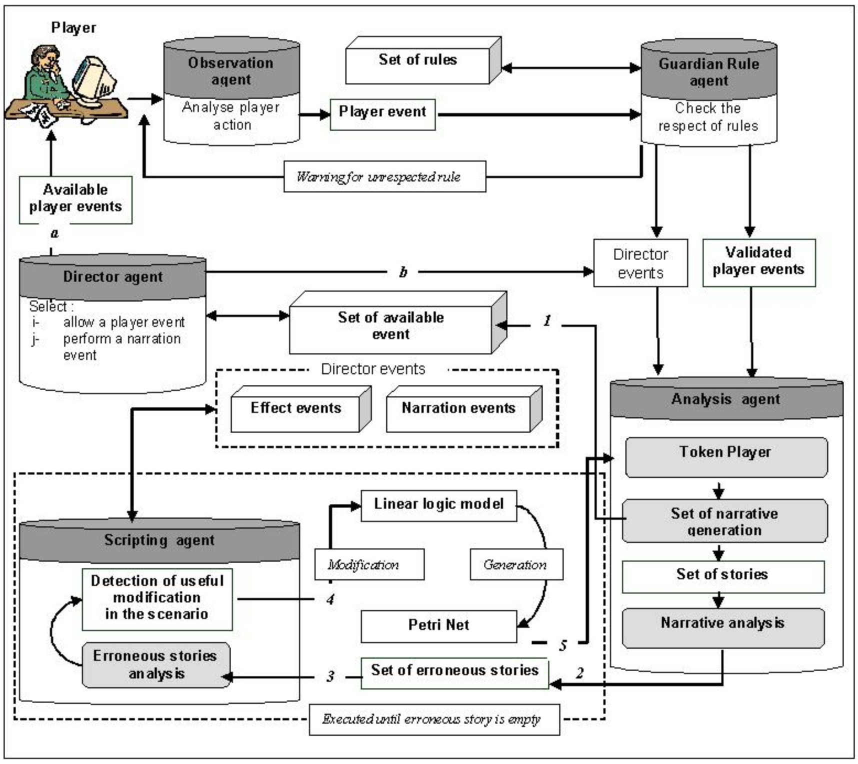


Figure 2: Scenario driving architecture

4 Properties validation

This section presents a set of properties that should be verified in one hand during the *game design* part and in other hand during the unfolding of the narration, called *execution properties*. Finally, we give some elements about validation of those properties.

4.1 Properties definition

As described in the previous section, the *analysis agent* is in charge to compute a set erroneous stories from the formal model and with respect to the following set of properties. The former class of properties is a mapping of classical properties on game interaction narration analysis. This answers the questions, what is safety for games? What is liveness when unfolding a narration? etc.

The class of properties verified during the game execution concerns a specific execution relative to specific behaviour of a player. They are divided in two class, *control* properties and *behavioural* properties.

Behavioural properties concern the *game unfolding* and it's *drama intensity*. Those ones are specific and depend from the context of interactive narration.

Before describing these properties it has to be said that in game narration, unlike in embedded systems, the path is more important than the result. As consequence properties analysis is not only it is correct or no, it is correct or no AND with the demonstration.

4.1.1 Classical properties

This first set of properties has been determined by classical approaches. In the context of game narration we have adjusted safety and livness properties.

- **Safety**. In computer validation it is defined as a property that express an undesirable event cannot occur. For example, in an exploration game the state of game where the player cannot progress any more.
- **Liveness**. A desirable state can be reach as often as required. In exploration game it corresponds to the player to be able to manage to change its weapon at each moment.
- **Reachability**. A specific state is reachable. The entire outcome defined by the game designer can be reached.
- **Deadlock freeness**. The game enters in a state where it cannot progress any more. For instance, the player enters a place and cannot find any resources for progressing.

These classical properties are partial regarding the properties of games. For instance it is impossible to determine if the story is "good" or if it is complex enough, etc.

4.1.2 Execution properties

Game unfolding properties are based on the story. It should be pointed out that the unfolding of a narration (building the story) is made by the user. Unlike control system these properties will be used to constraint the user ability whereas in control system properties are used to derive the control. However a model for unfolding narration and automatic proof can be derived.

The aim of drama properties is to characterise tension of the game. This allows to quantify the immersive of the game. However, it is difficult to express and analyse such properties in a formal way. This depends on expert appraise.

Game unfolding This set of properties is very specific to the unfolding of the narrative. The objective is to determine if the story generated by player when playing corresponds to the game designer requirements.

- **Symmetry**. It concerns the initial starting conditions for players. They start game in symmetrical conditions, have access to the same resources and information.
- **Justice**. N players have the same possibility to win the game with respect to the resources disposition on the map. When analysing all the stories of the game, we have to guarantee that the set of winning sequences is equal for each player.
- **Sequencing**. Respect of the precedence constraints between events. For instance [Ela02] presents a part of a game where the player character meets the wizard and is given the quest; the player character follows the wizard's instructions, finds, battles and defeats the dragon, returns its head to the wizard, and is rewarded with the key; the player character can now continue in the game by seeking and entering the underground cave system to further higher level quest. But if the player defeats the dragon before he met the wizard, he cannot follow its quest.
- **Complexity**. In order that the scenario will be enough interesting for the player, the game execution have to be enough complex. For example, the game cannot ended if a specified number of execution steps have not been performed. Let us consider a player that can win a level after two or three steps in this level. This kind of game execution has to be avoid.
- **Loopholes**. Can be defined as a flaw in the system which users can exploit to gain an unfair or unintended advantage.
- **Dead-ends**. Occurs when a player gets stranded in the game and cannot continue towards the game objective no matter what they do. In a game where specific resources necessary to win the level, it can happens that the player forget to catch it. We have to guarantee this missing will not be fatal. And for example produce this resource later.
- **Completeness**. An internally complete game is one in which the players can operate the game without reaching any point at which either the gameplay or the functionality is compromised.

Classical and game unfolding properties allows to define the correctness of a narration. But the question for game is: is the game fun? We believe that increasing interest for game depends on the narration and feelings that comes from.

Drama of the story What makes the player expecting the next event? May be its willing to live an experiment, discover the end of the story. This implies that story generate drama tension. Then it has to be characterised.

- **Frequency of drama tension**. In the game unfolding the set of events that produce specific drama tension is performed to give the player the immersive sensation. These kind of events should appear frequently but with a certain distance.
- **Diversity of drama tension**. There are a lot of ways to make the player feel drama tension. For example the player can feel fear, stress, happiness, etc. When performing drama tension during the game course, the narrative has to mix this kind of tension in an equal balancing.
- **Balancing**. The balance between drama tension situations and calm situations.
- **Local outcomes**. The game is not only defined by a final outcome, but also by local ones that he has to reach in order to progress until the end of the game.
- **Matching drama**. A game is the process of making sure the gamer meets the goals you've set for the player experience.

4.2 Analysis agent toolbox

This section presents our framework analysis and then gives some propects for analysing the described properties. Indeed, we present some elements on the *analysis agent* features.

As our purpose is to build an adaptative system that helps the user to derive a story, this agent will produce a set of erroneous stories depending on the context and previous choice of the player. Then, the *scripting and the director agent* will controls events the player can achieve in order to provide an adaptative sequence

Verifications during the game execution are based on **a state characterisation**. In each step of the game verifications are performed in order to establish the quality of the state reached by the game at this point of the story told to the player. The objective is to validate a set of properties at the current execution point and until the end of the game. We will give further tips in order to validate properties previously defined.

To conclude on properties definition, it can be said that classical and game unfolding properties can be computed and then directly included in the control of the unfolding of narration. On the opposite, drama can be defined in an accurate way but seems to be difficult to be determined in an automatic process.

Local and global validation The unfolding of a narrative is event based and it is difficult to model the player. So the control of the story depends on player actions. The scope of properties is then an important parameter for validation. In the sequel the analysis agent will perform two kind of validation's scope: local and global.

First local validation. At each occurrence of an event a set of properties should be validated. It should determine if properties are true at that time and for future. Local validation are based on a bounded future event occurrence.

Second global validation. This concerns properties based on the course of game and related to the end of the game (actually the end of the unfolding). Validation has to determine if a set of properties is valid for each possible stage of all the possible unfolding. The analysis agent gives a state characterisation of validation at each stage. The following paragraph details these state characterisation.

State characterisation The analysis agent performs a set of verification on the game unfolding. These verifications are based on the following states characterisation:

- *stable states*: states where all the properties are guaranteed to be verify until the end of the game.
- *semi-stable states*: states where all the properties are verified during a delimited window of executions steps.
- *unstablest states*: states where one of the story property is not verified in the defined window of execution step

The framework for properties validation has been introduced (adaptative system, scope of validation and state characterisation). We will, in the sequel, present approaches to perform validation of properties.

4.3 How validating these properties

Let us deal with validation perspectives. Local properties can be directly derived from a current stage of the narrative execution. We will focus on global properties.

Classical and game unfolding properties validation Classical properties and game unfolding properties will be validated with the same kind of method.

Safety verification consists in unfolding all the possible narratives from a starting point. Then determine if unexpected event can occur. Unfolding all the possible stories is not possible in reactive execution.

Reachability concerns a specific event in the story. The structure of such properties allows not deriving the whole possible sequences. Indeed, when the search event or state has been reached, the analysis agent will consider that the property is verified. The narrative is unfolded until a specific event appears.

Deadlock freeness must either analyse by unfolding all the possible narratives (like safety). In order to match reactive execution requirements it should be pointed out that it is also possible to perform an analysis on the structure of the model (such as Petri net analysis [ECS93, BA95]). It is then possible to modify the control model of the unfolding in order to be deadlock free.

Due to interactive narration requirements and state space explosion problem it is not possible to verify safety and deadlock freeness for stable states. It is only possible to state that a semi-stable state will respect safety properties during a specific window of events.

Validation of unfolding properties consists in generating all the reachable sequences of the game and analysing these sequences. The analysis of sequences is based on the comparison of the sequences with language description properties. The explosion problem is a limitation for interactive narration control. A possible approach is to apply abstraction techniques during the verification part like in [PCDR02]. Another solution is to performs local verifications to avoid the unrespect of game unfolding properties during a specific window of event and this either if we are not able to guarantee their respect until the end of the game.

Drama of the story The evaluation of the drama of the story needs to be able to keep trace of the preceding events that have appeared during the game execution. This will be called the **drama profile** and will contain some information about the tensions, and positive feelings that the player have felt until the beginning. It can contains for example, the number of PNJ killed by the player, the set of discover that he have made, the local outcomes he has performed, the time he need to reach this point of the game.

With respect to this profile it is possible to determine if the near future needs of the story have to become calmer or more stressing for the player. Indeed, as the game is driven by the adaptative system it is possible to change the future of the game. This will be performed by keeping the global outcomes and unfolding and the game but by adding or retrieve events for keeping a controlled drama.

At each point of the game unfolding, we will determine if this drama (described by the drama profile) is enough intense and if it is not the case, directives are performed to increase this drama. As it has been explained before, the directives are described by the designer of the game. The pre-conditions for triggering those directives must be satisfied in order that the adaptative system runs them.

5 Conclusion

This paper presents properties analysis benefits for interactive narration game. In such games the story is driven by the player action (interaction) but the challenge is to make feel the player what the designer of the game pretends to. In other terms, the game has to propose the player only actions that correspond to an bounded unfolding.

[CPE05] proposes a method and an architecture for the adaptative system responsible of the unfolding of the story. It is based on rewriting a model of narrative at each state. In this system the scripting agent manages the model, produces the set of erroneous stories and the director agent modifies the model in order to avoid erroneous stories.

One can argue that disabling and enabling events may change properties (mainly execution properties). This is true, but as previously said, we are focusing not only on the result of the properties but also on how it has been proved. Then a property is given for a certain frame. The weakness of the approach is that it cannot guarantee that the new model is better. Formal properties analysis will help in defining a controller that goes through a stable scenario.

A perspective of this paper is to formally define the set of properties by means of formal model (such as Linear Logic). And the challenge is to define methodologies for validation of game properties.

References

[aCSaSH04] T. Fullerton C. Swain S. Hoffman. *GAME DESIGN Workshop (Designin, Prototyping, and Playtesting Games*. Number ISBN 1-57820-222-1. CMP Books, 2004.

[AL04] R. Aylett and S. Louchart. Towards a narrative theory of virtual reality. *Virtual Reality, special issue on storytelling*, 7, 2004.

[BA95] K. Barkaoui and I. Ben Abdallah. Deadlock avoidance in fms based on structure theory of petri nets. In *Emerging Technologies and Factory Automation*, pages 499–510, Paris (France), October 1995.

[CCP05] F. Collé, R. Champagnat, and A. Prigent. Scenario analysis based on linear logic. In *ACE*, Valence (Spain), July 2005.

[CLM+03] F. Charles, M. Lozano, S. J. Mead, A. F. Bisquerra, and M. Cavazza. Planning formalisms and authoring in interactive storytelling. In *1st international conference on technologies for interactive di gital storytelling and entertainment*, Darmstadt (Germany), 2003.

[CPE05] R. Champagnat, A. Prigent, and P. Estraillier. Scenario building based on formal methods and adaptative execution. In *ISAGA*, Atlanta (USA), June 2005.

[ECS93] J. Ezpeleta, J.-M. Couvreur, and M. Silva. A new technique for finding a generating family of siphons, traps and st-components. application to colored petri nets. *LNCS: Advances in Petri nets*, 674:64–73, 1993.

[Ela02] M. Eladhari. Object oriented story construction in story driven computer gam es, 2002.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[GPNV02] V. Gal, C. Le Prado, S. Natkin, and L. Vega. Writing for video games. In *VRIC*, Laval (France), 2002.

[Jen03] H. Jenkins. Game design as narrative architecture. In N. Wardrip-Fruin and P. Harrigan, editors, *Fist person: new media as story, performance and game*, Cambridge, 2003. MIT Press.

[Juu99] J. Juul. A clash between game and narrative, February 1999.

[Juu04] J. Juul. Introduction to game time. *First Person: New Media as Story, Performance, and Game*, pages 131–142, 2004.

[LA04a] S. Louchart and R. Aylett. Emergent narrative, requirements and high-level architecture. In *SETN*, 2004.

[LA04b] S. Louchart and R. Aylett. The emergent narrative theoretical investigation. In *Narrative and learning environments conference NILE*, pages 24–33, Edinburgh (Scotland), 2004.

[LA04c] S. Louchart and R. Aylett. Narrative theories and emergent interactive narrative. *IJCEELL*, 2004.

[Lin02] C. Lindley. The gameplay gestalt, narrative, and interactive storytelling. In *Computer games and digital cultures*, Tampere (Finland), June 2002.

[Mat02] M. Mateas. *Interactive drama, art, and artificial intelligence*. Ph.d. thesis, Carnegie Mellon University (School of Computer Science), Pittsburg (USA), December 2002.

[ML03] B. Magerko and J. Laird. Building an interactive drama architecture. In *1st international conference on technologies for interactive di gital storytelling and entertainment*, Darmstadt (Germany), March 2003.

[Nat04] S. Natkin. *Jeux vidéos et médias du XXI siècle*. Ed. Vuirbert, Paris, 2004.

[PCDR02] A. Prigent, F. Cassez, P. Dhaussy, and O. Roux. Extending the translation from sdl to promela. In *9th International SPIN Workshop on Model Checking of Software*, pages 79–94, London (UK), 2002.

[RY03] M.O. Riedl and R.M. Young. Character-focused narrative planning. Submitted to Virtual Reality 2003, 2003.

[Szi99] N. Szilas. Interactive drama on computer: beyond linear narrative. In *AAAI Fall symposium on narrative intelligence*, pages 150–156, 1999.

[Vég04] L. Véga. *Petri net based model for analysing the event sequences in video games*. Phd thesis, CNAM, Paris (France), November 2004. (in French).

[VN03] L. Vega and S. Natkin. A petri net model for the analysis of the ordering of actions in computer games. In *X*, 2003.

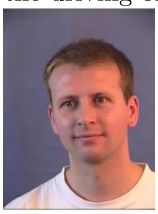
[You03a] R. Michael Young. Steps towards a computational theory of interactive narrative in virtual worlds. Unpublished document, 2003.

[You03b] R.M. Young. Story and discourse: a bipartite model of narrative generation in virtual worlds. Submitted to Virtual Reality 2003, 2003.

[YRB+03] R.M. Young, M.O. Riedl, M. Branly, A. Jhala, R.J. Martin, and C.J. Saretto. An architecture for integrating plan-based behavior generation with interactive game environments. Submitted for Publication in Journal of Game Development, vanlent@icts.usc.edu and niles@charlesriver.com, 2003.

Biography

After being lecturer from 2003 to 2005 at the university of La Rochelle, A. Prigent is now assistant professor at L3i laboratory. Armelle Prigent received her PhD in computer science in 2003. During her doctoral work, she proposed a method for the parameterized test of real time systems. Her research interests are on formal approaches for the driving of interactive application.



R. Champagnat is assistant professor at L3i laboratory since 2002. He receives a PhD degree in Petri net modelling of complex system. Its research interests are on interactive narration control and analysis by means of Linear Logic.


Pascal ESTRAILLIER is a full professor in the Computer Sciences department of the University of La Rochelle. His research concersn the architecture of software components in distributed and cooperative systems. He applies its results on Multiagent Paradigm and use formal specification theories in order to validate the behaviour and the interactions between components and to manage interoperability constraints.

He heads the L3i Research Laboratory Informatics, Image, Interactions to which 67 researchers are affiliated. He is Vice-President of the University of La Rochelle, in charge of development and industrial relationships. Moreover, He is, in shared time, project manager in the department Information Technologies of the French Ministry of research, in charge of Computer Science area.He is co-founder of the first French Educational program on Games at Master Level.