Linear Logic Validation and Hierarchical Modeling for Interactive Storytelling Control

Kim Dung Dang, Phuong Thao Pham, Ronan Champagnat, Mourad Rabah

Linear Logic Validation and Hierarchical Modeling for Interactive Storytelling Control

Kim Dung Dang, Phuong Thao Pham, Ronan Champagnat, and Mourad Rabah

University of La Rochelle - L3i, Avenue Michel Crépeau, 17042 La Rochelle, France
{kim_dung.dang, phuong_thao.pham, ronan.champagnat, mourad.rabah}@univ-lr.fr

Abstract. The games are typical interactive applications where the system has to react to user actions and behavior with respect to some predefined rules established by the designer. The storytelling allows the interactive system to unfold the scenario of the game story according to these inputs and constraints. In order to improve system's behavior, the scenario should be structured and the system's control should be validated. In this paper, we deal with these two issues. We first show how to validate Interactive Storytelling (IS) control using Linear Logic (LL). Then we present "situation-based" hierarchical scenario structuring which allows the state space reduction.

Keywords: Video game, Linear Logic, Interactive Storytelling, scenario validation, game controller, situation-based scenario.

1 Introduction

Video games, nowadays, are considered as one of the most popular media that has important contribution to the edutainment domain. Indeed, [6] demonstrated that, thanks to the effects of interactivity, video games brought a learning support that was more efficient than the traditional supports (non-interactive media formats such as text, oral presentation, video). However, the unfolding of a game (the unfolding of the story corresponding to the game) and its level of interactivity are commonly thought to be opposite [2]. The first relates to a designer's control on the game s/he has created as the second relates to a player's control on the game s/he is playing. In order to deal with this opposition, we have proposed an approach based on a LL model (an executable formal model) [1], which allows balancing these two controls.

The weakness of this approach is the explosion of state space for complex scenarios. Furthermore, a game designer cannot plan all the possible actions that a player can realize. To handle this issue, we propose to contextually structure the application execution into interaction sequences, called "situations". Each situation corresponds to a contextual resource-centered sequence of activities/events, and it is characterized by pre-conditions and post-conditions. That enables the system to control the execution and to establish casual links between the situations. Thus, the model confines actors' interactions according to shared contexts allowing hierarchical view of the scenario.

In this paper, we present briefly the usefulness of LL in modeling, validating and controlling an IS. We explain then the hierarchical situation-based modeling that we use to reduce the overall state space. We applied our approaches to build a video game: Little Red Cap (LRC) adventure game. In this game, the player plays the LRC character and the game controller adapts the scenario unfolding to her/his actions regarding game designer's desired effects.

2 Modeling, Validation and Control of Game Scenarios Using LL

In order to create interactive video games whose scenario (a set of all the possible discourses) satisfies the designer's intention, our approach is to model game scenarios by canonical narrative schemas introduced in Greimas' semiotics [1]. These schemas may be directly formalized by a LL model and thus we can validate game scenarios by proving the received LL model. This valid LL model is then used as the input of a game controller. The game controller aims to manage correctly the unfolding of the game by taking into account player's action choices and calculations executed on the LL model. Thanks to the modeling process, the game designer may determine the required goals or a structure of discourse that the game has to obtain. In other words, the LL model represents a predefined scenario of the game (interested readers can get more information in [1]). Thus, the game controller is able to follow the execution track, and hence will operate correctly: avoid incoherent states, guarantee the player's freedom and the consistency of the generated discourses.

Discussion: The scenario modeling by LL shows a lot of advantages. The game designer may balance between both the discourse point of view and the character point of view for the created games by taking into account both system's choices and player's choices thanks to additive connectors in LL [1]. Moreover, the modification of the modeled scenario is simple: we are able to make more or less discourses by adding or deleting events/actions, which allows increasing the adaptability of the scenario. However, this approach may lead to a state space explosion if there are a lot of events/actions in the story. Besides, the current LL approach does not use the interaction context, user's states, resource constraints... Finally, it is not really suitable for applications where interactions are totally unplanned. For these reasons, we propose the notion of situation that can be seen as a scene in a scenario. It encompasses not only interaction execution but also interaction management and resource use. The situations, as basic narrative elements, respect the overall scenario structure and facilitate interaction planning and management by characterizing, contextualizing and confining them.

3 Situation-Based Scenario

In general, in an interactive system, the execution is composed of a succession of activities and actions performed by the actors or a sequence of events affecting their behavior. Hence, a story is defined as a set of partially ordered events that solve the storytelling problem. That leads us to organize a scenario into a set of interaction sequences called “situations” [5].

3.1 Situation Model

The interactions are split into a set of situations. Each situation is a sequence of interactions between two or more actors in a precise context to achieve a predictive objective. It is characterized by: the pre-conditions, the post-conditions, a set of actors, and a set of resources. Due to the fact that actors' behaviors, especially human behaviors, are not always precisely modeled, and due to the influence of the external events, the progression of a situation can be considered as an “execution and adaptation black box” where the interactions are executed in a non-predictable way. To theses components we add a “consistency management component”: set of mechanisms devoted to prevention, detection and treatment solutions, to adjust the situation's progression in spite of misunderstanding and inconsistency problems.

3.2 Application Execution and Adaptation

The situations are considered as the plot structuring elementary blocks. Each application provides a set of situations defining all the possible interaction sequences that can happen during the application execution. They can be grouped and linked together to build the overall application scenario. The scenario is then represented by a directed graph of situations. Each node is a situation and each edge is a transition from one situation to another. The situation graph shows causal relationships between scenario situations, without taking into account used resources and event management. A given situation can be followed by multiple situations. A scenario may then have several beginnings and some possible endings. This allows expressing complex scenarios with a lot of paths. Moreover, the situations are defined regardless of the scenario conception and gathered in a situation library. The scenario definition by the application designer then consists in combining these available situations.

The advantage of the situation-based model against the LL approach and some existing story representation models [3, 4] is that it enhances the execution control and interaction adaptation. The application progression becomes the scenario unfolding from one starting node to one final node on the predefined situation graph. The drawback of this static chaining is its rigidity that does not allow the graph modification once the scenario execution begins. Therefore, in order to increase the adaptability, the second method is proposed: no predefined graph where situation choices are made according to the pre-conditions that best satisfy the global state and decision criteria. Thus, this method is more flexible, adaptive, and applicable in “real time” during the application execution.

4 Case Study: Little Red Cap Video Game

We have built a video game on the LRC story to which we have added multiple options for the non-player and player characters to increase the unpredictability as well as the interactivity of the game. The game scenario is modeled and validated by the LL approach to guarantee the objectives that the game has to reach. In order to reduce the state space explosion, we have also applied our situation-based structuring on the created scenario by reorganizing it with “situation” blocks. The starting point of this transformation comes from the fact that these two approaches are based on common notions such as events/actions, states, state transitions... Our idea is to reorganize the scenario graph by regrouping the events/actions, happening in a same context and relating to each other as a chain of different interactions, into one same situation. Therefore, we can eliminate the interaction sequences repeated in the graph obtained after the initial modeling for the LL validation. Thus, we are able to avoid the redundancy in the scenario graph. As a result, we have obtained a situation graph with 20 nodes instead of the initial graph with 56 nodes in the LL approach.

5 Conclusion

In this paper, we addressed the IS control from the validation and structuring points of view. We have tried to combine the user choices with the game designer’s intentions. To this purpose, we use a LL model to express the causality of the story and the distinction between player's choices and game controller's choices. LL is also well suited to prove the correct scenario execution and properties such as the reachability of predefined system states and the absence of deadlock.

However, the drawback of this approach is the state space explosion. In order to deal with this problem, we propose a hierarchical model based on the notion of “situation”. A “situation” encloses a set of interactions related to a shared context of several actors. In addition, each situation contains a set of management components that ensure a correct termination of the enclosed sequence. Compared to the state diagram, the situation can be seen as a higher level block representing several initial states and transitions. This approach allows us to reduce the state space by eliminating the redundancy in the interaction sequences in the modeled scenario.

References

1. Dang, K. D., Champagnat, R., and Augeraud, M.: Modeling of Interactive Storytelling and Validation of Scenario by Means of Linear Logic. In: Aylett, R. et al. (eds.) ICIDS 2010. LNCS, vol. 6432, pp. 153--164. Springer, Heidelberg (2010)
2. Juul, J.: A Clash Between Game and Narrative. In: Digital Arts and Culture Conference. Bergen (1998)
3. Magerko, B.: Story Representation and Interactive Drama. In: 1st Artificial Intelligence and Interactive Digital Entertainment Conference. Los Angeles, California (2005)
4. Mateas, M.: Interactive Drama, Art, and Artificial Intelligence. PhD Thesis, School of Computer Science, Carnegie Mellon University (2002)
5. Trillaud, F., Pham, P. T., Rabah, M., Estraillier, P., and Malki, J.: Online Distant Learning Using Situation-based Scenario. In: CSEDU2012 (2012)
6. Wong, W. L., Cuihua, S., Nocera, L., Carriazo, E., Tang, F., Bugga, S., Narayanan, H., Wang, H., and Ritterfield, U.: Serious video game effectiveness. In: Proceedings of the ACE 2007 conference. Salzburg (2007)