TOWARDS USING DEVS FOR MODELLING ADAPTIVE STORYTELLING IN VIRTUAL GAMES

Aznam Yacoub

Aix Marseille Université,
Université de Toulon, CNRS, LIS
MoFED
Marseille, France

Ecole Polytechnique de Montreal
HESL
Montreal, QC, Canada
aznam.yacoub@polymtl.ca

Gabriela Nicolescu

Ecole Polytechnique de Montreal,
Université de Montreal
HESL
Montreal, QC, Canada
gabriela.nicolescu@polymtl.ca

Maamar el-Amine Hamri

Aix Marseille Université,
Université de Toulon, CNRS, LIS
MoFED
Marseille, France
amine.hamri@lis-lab.fr

Claudia Frydman

Aix Marseille Université,
Université de Toulon, CNRS, LIS
MoFED
Marseille, France
claudia.frydman@lis-lab.fr

ABSTRACT

For the last decades, Verification and Validation techniques have been well improved in order to make safer complex systems. Sophisticated algorithms and methodologies have been proposed in the domain of formal modelling, and simulation. However, like for testing approaches, all these methodologies suffer from a strong weakness, while they depend on specifications, use cases and test scenarios. Indeed, if the experimental frame is not well-defined, modelling and simulation necessarily lack in accuracy. To overcome this challenge, a solution consists of automatically generate validated scenarios during the simulation, by considering the experimental frame also as a simulation model. Because this problem is similar to the adaptation of the stories in virtual worlds, this paper propose to explore a way to model adaptive storytelling in virtual games using the DEVS formalism.

Keywords: adaptive scenarios, virtual games, story generation, DEVS, interactive storytelling.

1 INTRODUCTION

Nowadays, systems integrate more and more different kinds of complex multimodal interactions, between the components which compose them, and also with the actors which interacts with them. The complexity of these interactions grows with the size of the global systems, which makes harder the study of their behaviours. Different strategies were defined in order to overcome this complexity, like the emergence of the field of Systems of Systems (SoS) Engineering (Ackoff 1971; Boardman and Sauser 2006; Nielsen et al. 2015). SoS break down the problem of complexity by considering that a system is a collection of individual and heterogeneous subsystems that act together in order to reach an overall goal, enhance the global robustness and increase the reliability of a complex monolithic system (Jamshidi 2008; Zeigler et al. 2016).

Furthermore, these systems usually need to be able to quickly adapt their behaviours to the conditions of the executing environment. For instance, multimedia and crossmedia applications like video games need to adapt themselves to the user profiles, in order to propose a better user experience (Streicher and Smeddinck 2016). In the case of emergency situations, systems need to adapt their actions to unexpected situations (Whittle et al. 2009; Esfahani and Malek 2013). However, Verification and Validation of the behaviours of such systems is harder while it is difficult to predict situations outside of the specifications. Especially in simulation, these assumptions impose the ability to infer new scenarios respecting the initial experimental frame, and bringing a real meaning for the system under study. In other words, verifiying and validating self-adaptive systems imply the ability of generating dynamically new adapted and validated scenarios.

From this statement, we can make a parallel between the problems addressed by the literature of Interactive Software and the literature of Verification, Validation and Test. While automated techniques for software test case generation has been intensively studied for the last decades (Lu 1994; Anand et al. 2013) the concept of self-adaptive scenarios is a main key in Digital Gaming (Rempulski et al. 2009; Chowdhury and Katchabaw 2013; Tregel et al. 2017), in which the story tries to fit to the user decisions in order to make it more attractive for the player. In Serious Gaming and Interactive Environment for Human Learning, adaptability is important while it enables the deployment of different approaches that match to the needs of each learner (Ismailović et al. 2012; El-Kechaï et al. 2015; Lavoue et al. 2018). In the case of Simulation of Self-adaptive Software (SaS), the problem of specifying and generating relevant scenarios to verify model adequacy and correctness is addressed (Muñoz-Fernández et al. 2015).

In short, scenario can independantly refer to stories, specifications, ordered steps or instructions, or ordered events. As a consequence, we assume that the problem of generating adaptive scenarios for the Verification and Validation of Simulation Models can be resolved by studying the problem from the point of view of *self-adaptive verified and validated storytelling*. Indeed, storytelling refers to the system that generates a story (i.e a scenario). Then, adaptive storytelling would be able to generate adaptive scenarios which acts on a self-adaptive model (the system under study). Then, if the self-adaptive model is coupled to the self-adaptive storytelling in a feedback loop, then the global model would evolve as a self-verified and validated self-adapted system.

Therefore, we propose in this paper, as preliminary work, a way to model adaptive storytelling by exploiting the hierarchical structure of Discrete-Event System Specifications (DEVS) in the case of Virtual Worlds and Digital Games. Section 2 makes a recall of definitions and related works about Adaptive and Interactive Storytelling (IS). Section 3 introduces an approach for modelling IS using DEVS. Finally, Section 4 focuses on a example of quest generation.

2 DEFINITIONS AND RELATED WORKS

2.1 Story and Storytelling

Storytelling is a concept that have been existing since the dawn of time as a kind of suspenseful knowledge transfer using exciting stories and immersion. From a conceptual point of view, it is a technique that summarizes past experiences by introducing a correspondance between words and sequences of ordered events

(Labov 1972). Andrews et al. (2009) says that a story is a mental structure of sequences of events according to different models (Hurme 2016).

For instance, the Plot Diagram defined by Aristotle (Figure 1) is divided into three narrative blocks, organized around five phase:

- Exposition introduces the scene, the characters and the situation of the story;
- Conflict introduces the dramatic narrative element which creates exciting story and introducing problems;
- Rising action is a sequence of consequences of the conflit;
- Climax is the peak of the consequences;
- Falling actions is a sequence of actions that resolve the problems;
- Resolution is the end phase of the story.

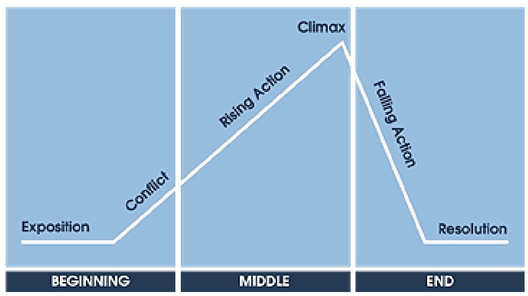

Figure 1: Aristotle's Plot Diagram.

Other story models follow a similar structure. Then, at a conceptual model, narrative is defined by two elements (Figure 2): a *Plot* (Forster 1927; Dibell 1999) is a macro-structure that organizes the timeline of events expressed by a story. Each component of a plot can be seen as a subplot that defines a coherent substory. A story is thus a micro-structure containing a sequence of ordered events called *narremes* (Dorfman 1969; Wittmann 1975). This means that the sequence of events is ordered both by the Plot and by the Story itself. Therefore, a narrative can be seen as a hierarchical structure of coherent stories.

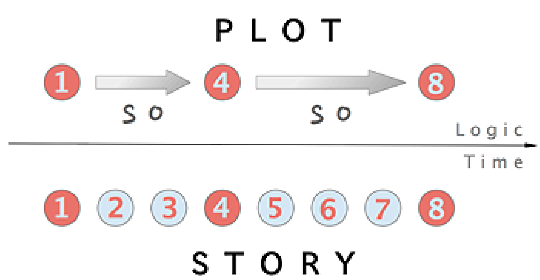

Figure 2: Plot and Narreme.

From this definition, we can deduce that a plot is a sequence of not necessarily ordered events, just a trace, while a story is a sequence of ordered events bound by causal relationships. Consequently, different models of plot can be defined (Göbel et al. 2009):

- linear plot generates linear stories where events are chronologically ordered. If the narrative medium (i.e the novel, movie, etc.) has multiple narratives, each of them are independant;
- non-linear plot (Figure 3) generates stories in which the events seem to be not bound by any chronological or causal relationship. In fact, inside each narreme, the events are well-ordered, but the subplots can be told in any orders. This means that events between subplots are not necessarily ordered;
- branching plot models the ability to generate different events in each subplot. The global story is thus an oriented graph, while the plot remains linear. This kind of narrative, employed in video games, gives the feeling that different stories exist inside a unique media, or that the story has multiple ends.
- interactive plot is a plot where the storyline is not defined. Only the settings (i.e the elements of the story like actors, characters, etc.) and possible narrative situations are described. In this kind of storytelling, the player creates its own story. The story adapts itself to the user.

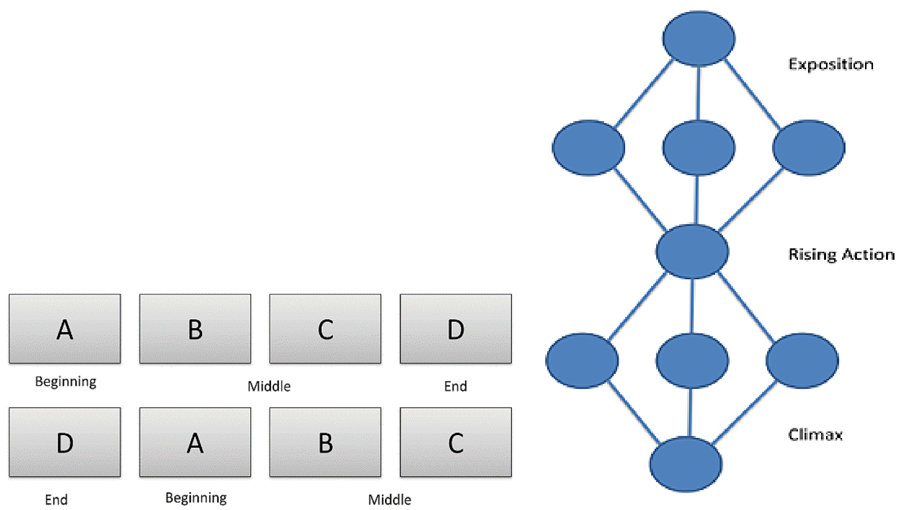

Figure 3: Non-linear and Branching Plot.

However, the main problem of interactive storytelling is the difficulty to validate the storyline, while it is likely impossible to check all the possible stories. Furthermore, the complexity grows with the size of the settings. Therefore, some restrictions need to be imposed in order to ensure the coherence of the generated stories.

2.2 Interactive Storytelling

Many approaches have been explored in the literature of interactive and adaptive storytelling. Firstly, multi-agent approaches are based on a common architecture (Arinbjarnar et al. 2009; Bostan and Marsh 2012):

- The Drama Manager is responsible for guiding the narrative by executing the best story events in a coherent sequence and reconciling contradictory plots.
- The Agent Model handles the behaviours of the non-player characters according to the drama.
- The User Model keeps track of player choices.

In Plot-based approches (Magerko 2003), the plot is seen as a state transition model. Each scene in a plot are represented by a graph of desired states to be reached. Each state are connected using logical clauses. In order to generate a story, the Director compares the behaviour of the user with the valid path written by the storyteller. A predicted user behaviour model completes the user model in order to prevent errors. If an error is detected, for instance that a path cannot be reached from a given action of the user, some decisions are performed to change the world or to guide the user to an action that leads to a path of a the planned story. The main limitation is that the plot must be written, meaning the system does not really adapt to the user actions.

Rempulski et al. (2009) go further by introducing model-checking to drive and validate the storytelling. In this approach, the plot is not a transition-state model but a set of two controlled automata: the scenario that represents the set of the game's entities and their possible evolutions, and a decision model that represents the decision of theses entities. Adaptation is then made by chosen a new path in the controlled scenario model according to the events generated by the actions of the user. Some paremeters like drama intensity, difficulties or challenges can be thus changed.

In Character-Based approaches (Cavazza et al. 2002), the actions of each agent in the story are planned using a Hierarchical Task Network (HTN). Each HTN represents several decompositions for the main task given by the Drama Manager. Then, the decision making is done by two processes. On the one hand, the graph resulting from the interleaving of all the HTNs is explored depth-first. An heuristic function evaluates the score of each possibility to maximize the score of the goal. On the other hand, backtracking helps to resolve narrative relevance with two mechanisms: situated reasoning and action repair (Paul et al. 2011). Situated reasoning allows obtaining a specific resulting state in a given situation. Action repair consists of finding a new state if the current one lead to a path that does not satisify the executability conditions. If this methodology allows generating coherent stories, the problem is that it does not take into account intertwined plots and dependancies between actions, which greatly increase the complexity. Storylines need thus to be simple, and users are always seen as observers.

We propose in this paper to extend these works by using DEVS for modelling interactive storytelling. Indeed, from the definitions given previously, a story can be seen as a hierarchical event-oriented model. Therefore, using DEVS (Zeigler 1976; Zeigler et al. 2000) for modelling stories seems to be natural, while simulation can help the storytelling to adapt the story progressively to the evolution of the virtual world. In fact, we assume that it is possible to replace an entire subplot by another one using this technique as we see in the next section. Moreover, quantitative aspect of time can be also taken into account, unlike the existing approaches which are interested only in order of events.

3 MODELLING THE STORYTELLING USING DEVS

3.1 Story Modelling and Architecture

Our proposed architecture consists of an hybrid approach based on hierarchical layers. At the macro-level, an approach similar to the plot-based one is applied in order to make the plot evolving according to a storyline. At the micro-level, planning is used to reach the goal of each subplot. As a consequence, it is possible to change the story in three manners:

1. Changing the settings. We mean replacing the actors or the objectives by equivalent ones at the micro-level.
2. Changing the structure at the macro-level. We mean replacing a subplot by another one or dynamically adding/removing subplots.
3. Changing the storyline. We mean generating new acceptable conditions of execution.

The two first modifications can be achieved thanks to the DEVS architecture, while the third one can be achieved using combined model-checking and simulation (Yacoub et al. 2017; Zeigler et al. 2017). The structure of the story is then defined as follow.

Definition 1. *We call storyline, a logical formula that the story must verify at each step. The logical formula expresses a set of event that must be generated by the storytelling.*

Definition 2. *A plot is thus any sequence of events, a trace, that fulfills the storyline. A narreme is thus by definition a plot. While the story is modelled using DEVS, a narreme is a DEVS coupled model*

$$M = < X, Y, EIC, EOC, IC, SELECT >$$

where

- *X is the set of input events. These events come from the external environment and models the actions of the user;*
- *Y is the set of story events. The sequence of external events emitted by the system; it must match the storyline;*
- *EIC and EOC represents the coupling between a subplot and its parent;*
- *IC represent the coupling between the narreme inside a plot.*
- *$SELECT$ allows priority handling when two subplots are resolved in the same time.*

Definition 3. *At the micro-level, a scene is a collection of entities. An entity is a DEVS model A representing the settings. For instance, A can be a model of a character, objects, actions, etc. As a consequence, an actor in a scene can be instantiated multiple times in a story. Indeed, an actor can act in multiple narreme in the same time.*

While we do not enforce the model representing the entities, any formalism can be used in order to model the actors. The only condition is that they implement the DEVS-Bus (Zeigler et al. 2000) or can be encompassed in DEVS models.

It is important to understand the main difference with other approaches: an entity can evolve in several plots in the same time, while a plot does not necessarily corresponds to a scene.

To illustrate this approach, consider the following narremes:

1. A - Begin of the story;
2. B - The good knight must deliver the princess;
3. C - At the same time, the bad knight must deliver the princess;
4. D - End of the story.

Consider the following requirement: "After some adventures, the princess must be delivered. She will finish by enjoying life with its savior".

Then, we define the storyline as

$$A \implies (-) \implies D$$

where "$-$" represents any events.

The underlying coupled DEVS model is given in Figure 4. A, B, C and D represent scenes in which live DEVS models of the good knight, the bad knight and the princess. The green arrows mean that when the plot

A is finished, an event is emitted to active the intermediate plot, and when the intermediate plot is finished, the final plot is enabled. This enforces the linearity of the plot.

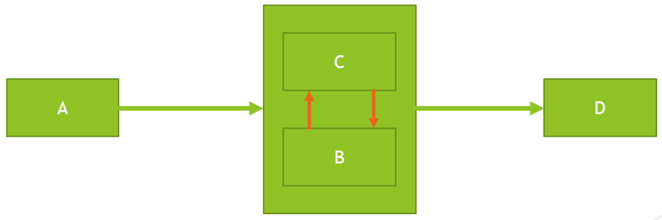

Figure 4: Example of the Princess and its Saviour.

In the same manner, narremes B and C can have interactions between them, while the actors that they involve can evolve in different scenes in the same time. Indeed, we mean that, if the bad knight saves the princess, the good knight can not save her anymore for example.

Narreme B can be itself refined in a DEVS coupled model (Figure 5) representing another branching subplot:

1. BA1 - The good knight must find Excalibur;
2. BB1 - The good knight must go the castle;
3. BC1 - The good knight must kill the big dragon;
4. BD1 - The good knight must open the door of the cell.

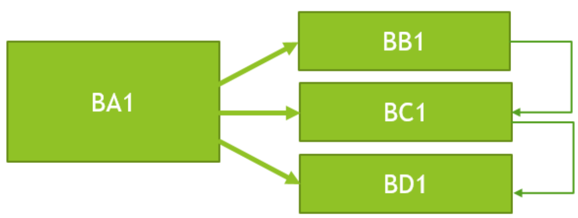

Figure 5: Example of the good Knight and how saving the Princess.

Then, what happends if Excalibur is found by another player ? In fact, while the scene is composed by each entity, if Excalibur is find in the narreme C, an event is also emitted in B by Excalibur itself through the coupling. This means that BA1 is automatically cancelled, and the narreme B cannot fullfilled its storyline. Depending on the structure of the storyline, we can then adapt the story by acting on the part that does not respect the requirement. For example, a possible strategy would be to replace $BB1$ by "Kill the bad knight which has Excalibur". Replacement can also be done using a semantic evaluation of the plot. For instance, we can classify the entities by types and replace them with their equivalent. However, this point is out of the scope of this paper.

This kind of architecture have a major drawback: while the actors can act independantly in any plots, we need to instantiate them several times. This leads to an explosion of the statespace. One solution would consist on using shared instances of DEVS model (Dalle et al. 2008). Another solution consists of outsourcing the scenario. By uncoupling the logical model of the game, and the storytelling, we allow reusability and more adaptability.

3.2 Outsourcing the Scenario

In this approach, we consider that the storytelling and the virtual worlds are two differents systems coupled in a feedback-loop. Then, agent model and user model are encompassed into a DEVS coupled model which is coupled with the storytelling previously defined. Similar to the plot-based approach, when an action is performed by the system, an event is emitted to update the story. Conversely, a change in the storytelling emit an event to the world model for enabling or disabling elements. In short, storytelling acts more like a monitor of the specifications of the game.

However, modifying the structure of the world for generating new contents is harder while removing an actor would need to explore and check the entire statespace of the storytelling. This could be impossible by considering timed events (while the story is a DEVS model, it can integrate complex timed relation between plots, events and entities).

4 APPLICATION : THE GIRL IN THE HOUSE

In order to validate our approach, we develop a prototype of adaptive scenario in a video game. The sequence of quests is written using a tool (Figure 6), and then translated into a DEVS coupled model. While we are just focusing on the hierarchical structure, we do not provide the implementation of all the entities as DEVS models (i.e. characters, items, etc.).

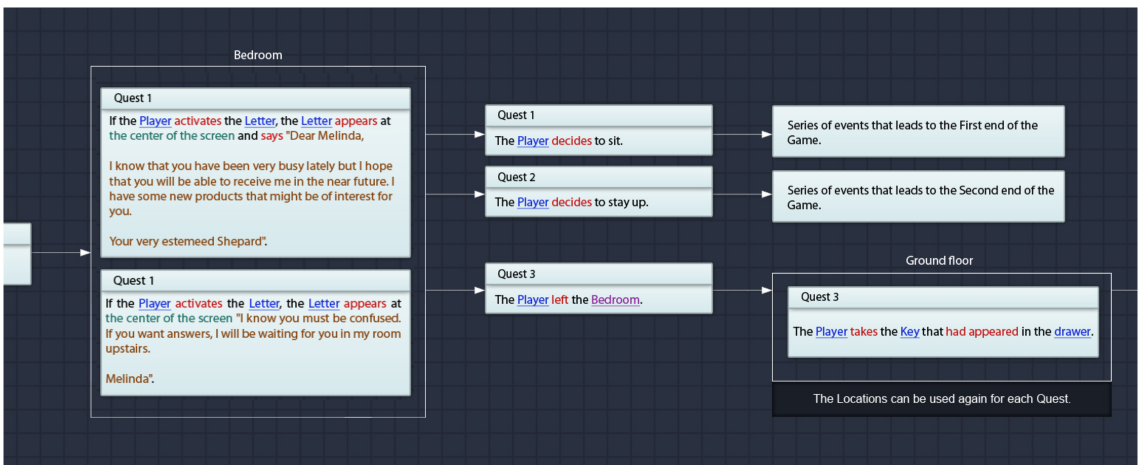

Figure 6: Example of Quest Writer.

In this example, the plot is divided into three rooms: the kitchen, the bedroom and the hall. The player has the possibilty in the hall to read two messages and to navigate freely in the house. A key lays into a drawer. One of the message enables the ability to speak to a character which asks the player to get the key in the drawer.

Modelling the narremes gives these steps and the underlying DEVS coupled model (Figure 7):

- A1 - The Player reads the message 1.
- A2 - The Player reads the message 2.
- B1 - The Player goes to the bedroom.
- C1 - The Player speaks to the the character.
- D1 - The character asks the player to pick the key.
- E1 - The player picks the key.

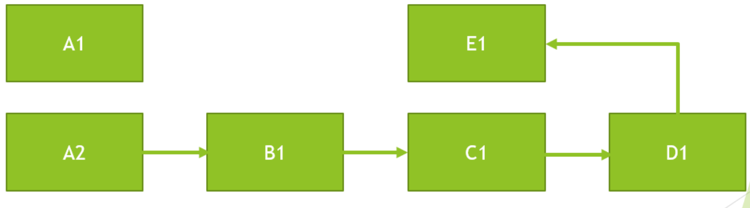

Figure 7: DEVS Coupled Model of the Girl in the House.

We define the storyline as :

$$A1 \vee (A2 \implies B1)$$
$$B1 \implies C1$$
$$(C1 \wedge \neg key) \implies D1$$
$$D1 \implies (E1 \wedge key)$$

What happends if the player picks the key before speaking to the character ? In fact, readers must remember all these plots occur in the same scene. Each entity is instantiated in each DEVS plot model. Then, when the key is taken by the player, an event is immediately emitted to C1 at this point of the story through the DEVS mechanism. While in C1 the condition to enable D1 is to have no key, simulation allows detecting that the storyline would not be met, because E1 was emitted before C1. Then, when E1 is emitted, one repair strategy consists of replacing the key by any equivalent objet in the storyline. Therefore, D1 is dynamically replaced by D1+ : "The character asks the player to pick the key 2", and E1 by E1+: "The character picks the key 2". This is done by just replacing the model of the key by another one in the DEVS structure of the scene. The storyline becomes:

$$A1 \vee (A2 \implies B1)$$
$$B1 \implies C1$$
$$(C1 \wedge \neg key2) \implies D1+$$
$$D1+ \implies (E1 + \wedge key2)$$

If we use the shared instance architecture, replacing the key involved in D1 by the key involved in D1+ also resolves all the dependencies. The scenarios is automatically fully adapted to the new story.

5 CONCLUSION

In this paper, we propose some preliminary works to model and simulate adaptive stories in virtual worlds using the DEVS formalism. This allows flexibility while the hierarchical structure allows splitting a scenario into small units which can be semantically evaluated. Elements of the scene are modelled using atomic DEVS model, inside each story unit. Therefore, when a path cannot be reached, a story unit can be dynamically replaced by another equivalent one in order to generate a new path. Therefore, this allows the played scenario to differ from the initial one. By coupling this methodology with a combined formalism (Yacoub et al. 2017), validating the new scenario ensures that the story stays coherent with the inital one. Time is also taken into account: the scenario is not just only a set of qualitative events, but also quantitative.

We also suggest that the storytelling can be considered as an independant system which is coupled with the game in a loop: the generated scenario influences the game which in turn influences the scenario. This allows us to consider the storytelling as *an experimental frame generator* for a virtual world. The logic of the story (the scenario as a set of output ordered events) is outsourced from the logic of the world (the modelled system). In this way, we define a manner to generate dynamically a new experimental frame from an existing one, and from the system itself. While the scenario generator is itself a DEVS model, a similar reasonning can be applied on it.

Future works concerns the study of the complexity of the generated DEVS model. Indeed, modelling a scenario can be considered from two different *execution contexts*: the model can be "plot-oriented", or "scene-oriented". In both cases, some redundant models need to be introduced for modelling the entities which can act in parallel plots or in parallel scenes. In these cases, the use of shared instances can lead to some situations in which the feedback loop prohibition is not respected. Finding a way of modelling this context in DEVS is interesting in order to reduce the size of the generated statespace. Furthurmore, we need also study the use of Dynamic-Structure DEVS for adding and removing plots and entities. Validation using simulation using this approach must also be addressed. Others questions concern how performing the semantical evaluation of a story unit, and how using machine learning in order to take into account parameters which come from outside of the system (user experience, etc.). Indeed, the replacement is currently done by making classes of equivalent entities, which needs the definition of an ontology to keep the meaning of a story. This adds a level of validation while the semantics can be changed over the discovery of new scenarios.

ACKNOWLEDGMENTS

Great thanks to Pr. Bernard Zeigler for its advices about this work. Great thanks also to Delphine Routin and Sylvain Roux for their help in the design of the case study.

REFERENCES

Ackoff, R. L. 1971. “Towards a system of systems concepts”. *Management science* vol. 17 (11), pp. 661–671.

Anand, S., E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. Grieskamp, M. Harman, M. J. Harrold, P. Mcminn, A. Bertolino et al. 2013. “An orchestrated survey of methodologies for automated software test case generation”. *Journal of Systems and Software* vol. 86 (8), pp. 1978–2001.

Andrews, D. H., T. D. Hull, and J. A. Donahue. 2009. “Storytelling as an instructional method: Definitions and research questions”. *Interdisciplinary Journal of Problem-based Learning* vol. 3 (2), pp. 3.

Arinbjarnar, M., H. Barber, and D. Kudenko. 2009. “A critical review of interactive drama systems”. In *AISB 2009 Symposium. AI & Games, Edinburgh*.

Boardman, J., and B. Sauser. 2006. “System of Systems-the meaning of of”. In *System of Systems Engineering, 2006 IEEE/SMC International Conference on*, pp. 118–123. IEEE.

Bostan, B., and T. Marsh. 2012. “Fundamentals of interactive storytelling”. *AJIT-e* vol. 3 (8), pp. 19.

Cavazza, M., F. Charles, and S. J. Mead. 2002, Jul. “Character-based interactive storytelling”. *IEEE Intelligent Systems* vol. 17 (4), pp. 17–24.

Chowdhury, M. I., and M. Katchabaw. 2013. “A Software Design Pattern Based Approach to Adaptive Video Games”. In *ADAPTIVE 2013, the Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications*, pp. 40–47. Citeseer.

Dalle, O., B. P. Zeigler, and G. A. Wainer. 2008, Dec. “Extending DEVS to support multiple occurrence in component-based simulation”. In *2008 Winter Simulation Conference*, pp. 933–941.

Dibell, A. 1999. *Elements of Fiction Writing - Plot*. Elements of Fiction Writing. F+W Media.

Dorfman, E. 1969. *An Introduction to Narrative Structures*. University of Toronto Press.

El-Kechaï, N., J. Melero, and J.-M. Labat. 2015, June. "Adaptation de serious games selon la stratégie choisie par l'enseignant : approche fondée sur la Competence-based Knowledge Space Theory". In *7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2015)*, edited by C. C. D. M. e. L. O. Sébastien George, Gaëlle Molinari, pp. 294–305. Agadir, Morocco. Conférence EIAH 2015.

Esfahani, N., and S. Malek. 2013. *Uncertainty in Self-Adaptive Software Systems*, pp. 214–238. Berlin, Heidelberg, Springer Berlin Heidelberg.

Forster, E. 1927. *Aspects of the Novel*. RosettaBooks.

Göbel, S., F. Mehm, S. Radke, and R. Steinmetz. 2009. "80days: Adaptive digital storytelling for digital educational games". In *Proceedings of the 2nd international workshop on Story-Telling and Educational Games (STEG'09)*, Volume 498.

Hurme, Jarkko 2016. "Storytelling in Video Game : Creating a Narrative for management game".

Ismailović, D., J. Haladjian, B. Köhler, D. Pagano, and B. Brügge. 2012. "Adaptive Serious Game Development". In *Proceedings of the Second International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques*, GAS '12, pp. 23–26, IEEE Press.

Jamshidi, M. 2008, May. "System of systems engineering - New challenges for the 21st century". *IEEE Aerospace and Electronic Systems Magazine* vol. 23 (5), pp. 4–19.

Labov, W. 1972. "The Transformation of Experience in Narrative Syntax.". [06.02.2002] Inst. f. allg. und angewandte Sprachwissenschaft/Abt. PhASI <18/294>.

Lavoue, E., B. Monterrat, M. Desmarais, and S. George. 2018. "Adaptive Gamification for Learning Environments". *IEEE Transactions on Learning Technologies*.

Lu, P. 1994. "Test Case Generation for Specification-based Software Testing". In *Proceedings of the 1994 Conference of the Centre for Advanced Studies on Collaborative Research*, CASCON '94, IBM Press.

Magerko, B. 2003. "Building an interactive drama architecture". In *In First International Conference on Technologies for Interactive Digital Storytelling and Entertainment*.

Muñoz-Fernández, J. C., G. Tamura, I. Raicu, R. Mazo, and C. Salinesi. 2015, July. "REFAS: A PLE Approach for Simulation of Self-Adaptive Systems Requirements". In *SPLC 2015*, Volume 1 of *Proceedings 19th International Software Product Line Conference*, pp. 444. Nashville, United States, Vanderbilt University.

Nielsen, C. B., P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska. 2015, September. "Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions". *ACM Comput. Surv.* vol. 48 (2), pp. 18:1–18:41.

Paul, R., D. Charles, M. McNeill, and D. McSherry. 2011. "Adaptive Storytelling and Story Repair in a Dynamic Environment". In *Proceedings of the 4th International Conference on Interactive Digital Storytelling*, ICIDS'11, pp. 128–139. Berlin, Heidelberg, Springer-Verlag.

Rempulski, N., A. Prigent, P. Estraillier, V. Courboulay, and M. Perreira Da Silva. 2009, November. "Adaptive storytelling based on model-checking approaches". *IJIGS* vol. 5 (2), pp. 33.

Streicher, A., and J. D. Smeddinck. 2016. *Personalized and Adaptive Serious Games*, pp. 332–377. Cham, Springer International Publishing.

Tregel, T., J. Alef, S. Göbel, and R. Steinmetz. 2017, Oct. "Towards Multiplayer Content Online Adaptation using Player Roles and their Interactions". In *Proceedings of The 11th European Conference on Games Based Learning*, edited by D. J. G. Dr Maja Pivec, pp. 677 – 686. Graz, Austria, Academic Conferences and Publishing International Limited.

Whittle, J., P. Sawyer, N. Bencomo, B. H. C. Cheng, and J. M. Bruel. 2009, Aug. "RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems". In *2009 17th IEEE International Requirements Engineering Conference*, pp. 79–88.

Wittmann, H. 1975. "Théorie des narrèmes et algorithmes narratifs". *Poetics* vol. 4 (1), pp. 19 – 28.

Yacoub, A., M. E. A. Hamri, C. Frydman, C. Seo, and B. P. Zeigler. 2017. "DEv-PROMELA: an extension of PROMELA for the modelling, simulation and verification of discrete-event systems". *International Journal of Simulation and Process Modelling* vol. 12 (3-4), pp. 313–327.

Zeigler, B. P. 1976. *Theory of Modeling and Simulation*. John Wiley.

Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. 2nd ed. Academic Press, Inc.

Zeigler, B. P., J. J. Nutaro, and C. Seo. 2017. "Combining DEVS and model-checking: concepts and tools for integrating simulation and analysis". *International Journal of Simulation and Process Modelling* vol. 12 (1), pp. 2–15.

Zeigler, B. P., H. S. Sarjoughian, R. Duboz, and J.-C. Souli. 2016. *Guide to modeling and simulation of systems of systems*. Springer.

AUTHOR BIOGRAPHIES

AZNAM YACOUB is a Postdoctoral Researcher in the Computer Sciences Department of Aix-Marseille University and Polytechnique Montreal. He holds a PhD in Computer Sciences from Aix-Marseille University. His research interests lie in Software Engineering, Verification and Validation, Modelling and Simulation especially in Digital Gaming. His email address is aznam.yacoub@lis-lab.fr.

GABRIELA NICOLESCU obtained her PhD degree, in 2002, from INPG (Institut National Polytechnique de Grenoble) in France, with the award for Best Thesis in Microelectronics. She has been working at Ecole Polytechnique de Montréal (Canada) since august 2003, where she is a professor in the Computer and Software Engineering Department. Dr. Nicolescu's research interests are in the field of design methodologies, programming and security for systems with advanced technologies, such as 3D multi-processor systems-on-chip integrating liquid cooling and optical networks. She published five books and she is the author of more than hundred articles in journals, international conferences and of book chapters. Her email address is gabriela.nicolescu@polymtl.ca.

MAAMAR EL-AMINE HAMRI is an Associate Professor at Aix-Marseille University. He holds a Ph.D. in Computer Sciences form Aix-Marseille University. His research interests include Modelling and Simulation using DEVS and its extensions. His email address is amine.hamri@lis-lab.fr.

CLAUDIA FRYDMAN is a Professor at Aix-Marseille University. She holds a Ph.D. in Computer Sciences. Her research interests include Modelling and Simulation especially using DEVS. Her email address is claudia.frydman@lis-lab.fr.